



Macintosh Human Interface Guidelines

by Apple Computer, Inc.

Apple Technical Library





Macintosh Human Interface Guidelines



Addison-Wesley Publishing Company

Reading, Massachusetts Menlo Park, California New York
Don Mills, Ontario Wokingham, England Amsterdam Bonn
Sydney Singapore Tokyo Madrid San Juan
Paris Seoul Milan Mexico City Taipei

🍏 Apple Computer, Inc.
© 1992, Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc. Printed in the United States of America.

No licenses, express or implied, are granted with respect to any of the technology described in this book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple Macintosh computers.

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, APDA, AppleLink, AppleShare, AppleTalk, EtherTalk, HyperTalk, ImageWriter, LaserWriter, Macintosh, MultiFinder, and StyleWriter are trademarks of Apple Computer, Inc., registered in the United States and other countries.

BalloonHelp, BalloonWriter, Finder, PowerBook, QuickDraw, ResEdit, System 7, and TrueType are trademarks of Apple Computer, Inc.

Adobe Illustrator and PostScript are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

AGFA is a trademark of Agfa-Gevaert. FrameMaker is a registered trademark of Frame Technology Corporation.

Helvetica and Palatino are registered trademarks of Linotype Company.

HyperCard, MacDraw, MacPaint, and MacWrite are registered trademarks of Claris Corporation.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

Varityper is a registered trademark of Varityper, Inc.

Simultaneously published in the United States and Canada.

LIMITED WARRANTY ON MEDIA AND REPLACEMENT

ALL IMPLIED WARRANTIES ON THIS MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

ISBN 0-201-62216-5
1 2 3 4 5 6 7 8 9-BA-9695949392
First Printing, November 1992



The paper used in this book meets the EPA standards for recycled fiber.

Contents

Figures and Tables xiii

Preface

About This Book xxi

Who Should Read This Book xxi
What's New in Macintosh Human Interface From Apple xxii
About *Making It Macintosh* xxii
What's in This Book xxii
 The Basic Philosophy xxii
 The Interface Elements xxiii
 Appendixes xxiii
Visual Cues Used in This Book xxiii

Part 1

Fundamentals 1

Chapter 1

Human Interface Principles 3

The Human Interface Design Principles 4
 Metaphors 4
 Direct Manipulation 5
 See-and-Point 7
 Consistency 7
 WYSIWYG (What You See Is What You Get) 8
 User Control 9
 Feedback and Dialog 9
 Forgiveness 10
 Perceived Stability 11
 Aesthetic Integrity 11
 Modelessness 12
Additional Issues to Consider 13
 Knowledge of Your Audience 13
 Accessibility 14

Chapter 2

General Design Considerations 15

Worldwide Compatibility 16
 Cultural Values 17
 Resources 17

Language Differences	18
Text Display and Text Editing	19
Default Alignment of Interface Elements	20
Keyboards	22
Fonts	23
Universal Access	24
People With a Physical Disability	25
People With a Visual Disability	25
People With a Hearing Disability	26
People With a Speech or Language Disability	27
People With a Seizure Disorder	27
Collaborative Computing	27
Concern for Other Users	28
User Identification	28
Access Privileges	28
Passwords	29
Data Encryption for Security	30
Clear Communications	30
Displaying the Current State of Data	30
Communicating With Other Environments	31
Network Transparency	31

Chapter 3 Human Interface Design and the Development Process 33

Design Decisions	34
Features Inspired by Market Pressures	34
Feature Cascade	35
The 80 Percent Solution	35
Managing Complexity	35
Using Progressive Disclosure	35
Implementing Preferences	37
Extending the Interface	38
When to Go Beyond the Guidelines	38
Build on the Existing Interface	39
Don't Assign New Behaviors to Existing Objects	39
Create a New Interface Element Cautiously	40
Involving Users in the Design Process	41
Define Your Audience	41
Analyze Tasks	41
Build Prototypes	42
Observe Users	42
Ten Steps for Conducting a User Observation	43

Chapter 4 **Menus** 49

The Menu Bar	52
Menu Behavior	55
Menu Elements	58
Menu Item Names	58
Grouping Items in Menus	60
Menu Dividers	62
Standard Characters and Text Style in Menus	64
Checkmarks and Dashes in Menus	64
The Ellipsis Character in Menus	67
A Diamond Mark in the Application Menu	71
Avoid Nonstandard Marks in Menus	72
Text Styles in Menus	73
Toggled Menu Items	75
Scrolling Menus	78
Hierarchical Menus	79
Pop-Up Menus	82
Standard Pop-Up Menus	87
Type-In Pop-Up Menus	91
Tear-Off Menus and Palettes	92
Tear-Off Menus	93
Palettes	96
Standard Macintosh Menus	98
The Apple Menu	98
About	98
File Menu	99
New	99
Open	101
Close	102
Save	104
Save As	106
Revert	107
Page Setup...	108
Print...	108
Quit	109
The Edit Menu	109
The Clipboard	111
Undo/Redo	113
Cut	114
Copy	115
Paste	115
Clear	117

Select All	117
Show Clipboard/Hide Clipboard	117
Create Publisher...	117
Subscribe To...	118
Publisher/Subscriber Options...	118
The Font Menu	120
The Size Menu	122
The Style Menu	124
The Help Menu	125
The Keyboard Menu	125
The Application Menu	127
Keyboard Equivalents	128

Chapter 5 Windows 131

Window Appearance	134
Document Window Controls	134
Use of Color in Windows	135
Utility Windows	137
Window Behaviors	139
The Active Window	139
Opening Windows	141
Window Display Order	143
Window Positions	146
The Default Position on a Single Screen	147
The Default Position on Multiple Screens	148
Dialog Box and Alert Box Positions	150
Closing a Window	152
Moving a Window	154
Changing the Size of a Window	156
Scrolling a Window	158
Scroll Bars	158
Scrolling With the Scroll Arrows	163
Scrolling With the Gray Area	164
Scrolling by Dragging the Scroll Box	164
Automatic Scrolling	166
The Zoom Box and Window Behavior	168
Splitting a Window	170
Window Pane Behavior	172
One Split per Orientation	173

Chapter 6

Dialog Boxes 175

Modeless Dialog Boxes	178
Modeless Dialog Box Appearance	179
Modeless Dialog Box Behaviors	181
Menu Bar Access	181
Accepting Changes in a Modeless Dialog Box	182
Completing Commands	184
Movable Modal Dialog Boxes	185
Movable Modal Dialog Box Appearance	186
Movable Modal Dialog Box Behaviors	187
Menu Bar Access	187
Modal Dialog Boxes	188
Modal Dialog Box Appearance	190
Modal Dialog Box Behaviors	191
Menu Bar Access	191
Stacking Modal Dialog Boxes	192
Alert Boxes	193
Alert Box Appearance	194
Note Alert Boxes	194
Caution Alert Boxes	195
Stop Alert Boxes	196
Basic Dialog Box Layout	196
Keyboard Navigation in Dialog Boxes	198
Dialog Box Messages	199
Standard File Dialog Boxes	200
Save Changes Alert Box	201

Chapter 7

Controls 203

Standard Toolbox Controls	204
Buttons	204
Button Behavior	205
Button Names	206
Radio Buttons	210
Checkboxes	211
Controls Not Supported by the Macintosh Toolbox	214
Sliders	214
Little Arrows	216
Outline Triangles	218
Other Elements for User Interaction	218
Text Entry Fields	219
Scrolling Lists	220

Chapter 8

Icons 223

Why Icons Work	224
Limitations of Icons	227
Designing Effective Icons	229
Use Appropriate Metaphors	229
Think About Worldwide Compatibility	230
Avoid Text in Icons	230
Design for the Macintosh Display	231
Use a Consistent Light Source	232
Optimize for Your Target Display	232
Maintain a Consistent Visual Appearance in an Icon Family	233
Use Icon Elements Consistently	233
The Finder Icon Family	234
An Icon Design Process	236
Black-and-White Icons	238
Color Icons	238
Icon Colors	240
The Apple Icon Color Set	240
Degradation of the Color Set Across Monitors	241
Selection Mechanism for Color Icons	241
Color Labeling Mechanism for Color Icons	242
Anti-Aliasing	243
Small Icons	244
Default and Custom Icons	245
Application Icons	246
Document Icons	247
Stationery Pad Icons	248
Query Document Icons	249
Edition Icons	250
Preferences Icons	250
Extension Icons	250
Control Panel Icons	251
Movable Resource Icons	252
Keyboard Icons	252

Chapter 9

Color 257

Color Design of Standard Interface Elements	258
Windows and Dialog Boxes	258
Menus	260
Pointers	260
Highlighting and Selection	260

Color Application Guidelines	261
Match Complexity to the Level of User	261
Design for the Macintosh	262
Design for Black and White First	263
Limit the Number of Colors	264
Colors on Gray	265
Beware of Blue	265
Small Objects	265
Color for Categorizing Information	265

Chapter 10

Behaviors 267

The Pointing Device	268
Mouse Actions	271
Clicking	271
Double-Clicking	272
Pressing	273
Dragging	274
The Keyboard	275
Character Keys	275
Enter	275
Tab	276
Return	276
Delete (or Backspace)	277
Clear	277
Escape	277
Modifier Keys	278
Shift	278
Caps Lock	279
Option	279
Command	280
Control	280
Type-Ahead and Auto-Repeat	280
International Keyboards	281
Arrow Keys	281
Appropriate Uses for the Arrow Keys	281
Moving the Insertion Point	282
Moving the Insertion Point in Empty Documents	282
Using Modifier Keys With Arrow Keys	282
Function Keys	284
Help	285
Forward Delete (Del)	285
Home	285
End	285
Page Up	286
Page Down	286

Selecting	286
Selection Methods	288
Selection by Clicking	289
Selection by Dragging	289
Changing a Selection With Shift-Click	289
Changing a Selection With Command-Click	291
Selections in Text	292
Selecting With the Mouse	293
Selecting Ranges	294
Selecting With the Arrow Keys	295
Selections in Graphics	297
Selections in Arrays and Tables	298
Editing Text	300
Inserting Text	300
Deleting Text	300
Replacing a Selection	301
Intelligent Cut and Paste	301
Editing Fields	302

Chapter 11 Language 305

Style	306
Terminology	307
Developer Terms and User Terms	307
Terms That Are Often Misused	308
Click	308
Checkbox	308
Document	308
File	308
Utility Window	309
Labels for Interface Elements	309
Dialog Box Messages	310
User Documentation	313
Online Help Systems	314
Provide Concurrent Help	314
Provide Multiple Levels of Help	314
Assist Users by Answering Their Questions	315
Keep the Help System Simple	316
Design Online Help as an Interactive Coach	316
Balloon Help	316
When to Use a Help Balloon	317
How to Write a Balloon	318
Wording for Specific Balloon Types	319
Buttons With Words	319
Menu Titles	320
Menu Items	320

Radio Buttons	321
Checkboxes	321
Groups of Checkboxes or Radio Buttons	322
Tools in Palettes	323
Window Parts	324
Modal Dialog Box on the Screen	324
Icons	324
Text Entry Boxes	325

Appendixes 327

Appendix A Resources 329

Association for Computing Machinery (ACM)	329
<i>Communications of the ACM</i>	329
SIGCHI	329
SIGGRAPH	330
CSCW	330
Human Factors Society	331
Human Factors Society Annual Meeting	331
<i>Human Factors</i>	331
<i>Human Factors Society Bulletin</i>	332
Apple Developer Information	332
APDA	332
Developer Support Center	333
In-House Development Support	333
<i>develop</i>	333

Appendix B Bibliography 335

Animation	336
Cognitive Psychology and Human Factors	336
Color	337
Environmental Design	338
Graphic and Information Design	339
Graphic Design and Drawing	339
Icons and Symbols	339
Typography	340
History of Human Interface	340
Human-Computer Design	341
Consistency	341
Direct Manipulation	341

Menus	342
Metaphors	342
Product Design	343
Usability Testing	343
User-Centered Design	344
Human-Computer Interaction	344
Language	346
Programming	346
Special Applications	347
Collaborative Computing	347
Hypertext	347
Multimedia	348
Online Documentation and Online Help	348
Universal Access	349
Visual Thinking	349
Worldwide Software	350

Appendix C Checklist 351

General Considerations	351
Graphic Design	353
Color	353
Icons	354
Windows	354
Dialog Boxes	355
Alert Boxes	357
Scrolling	357
Menus	358
Pop-Up Menus	359
Palettes and Tear-Off Menus	359
Mouse Standards	360
Text	360
Balloon Help	360
Keyboard Equivalents	361
Edition Manager	361
Documentation	362

Glossary 363

Index 373

Figures and Tables

Chapter 1	Human Interface Principles	3
	Figure 1-1	Direct manipulation 6
	Figure 1-2	An example of a bad message and an example of a helpful message 10
	Figure 1-3	Don't use arbitrary graphic elements 12
Chapter 2	General Design Considerations	15
	Figure 2-1	Menu bars in different languages 18
	Figure 2-2	English and Arabic dialog boxes 20
	Figure 2-3	Dialog boxes with display rectangles that are different sizes and the same size 21
	Figure 2-4	Right-to-left alignment of dialog box items 21
	Figure 2-5	The Keyboard menu 23
	Figure 2-6	The boundaries of a font 24
	Figure 2-7	The Sound control panel 26
	Figure 2-8	A shutdown message 28
	Figure 2-9	The AppleShare connect dialog box 29
Chapter 3	Human Interface Design and the Development Process	33
	Figure 3-1	An expanding dialog box 36
	Figure 3-2	Directions a window can expand 37
	Figure 3-3	An incorrect subpalette indicator 40
	Figure 3-4	A better subpalette indicator 40
Chapter 4	Menus	49
	Figure 4-1	The standard order of actions 51
	Figure 4-2	A menu bar 53
	Figure 4-3	Three menu bars 53
	Figure 4-4	The Finder menu bar in six languages 54
	Figure 4-5	An unavailable menu 55
	Figure 4-6	Opening a menu 56
	Figure 4-7	A feedback technique 57
	Figure 4-8	A typical menu 58
	Figure 4-9	A menu with adjectives 59
	Figure 4-10	Command names properly capitalized 59
	Figure 4-11	Unavailable items aren't highlighted 60
	Figure 4-12	Menus with appropriate groups 61
	Figure 4-13	Grouping items in menus 62
	Figure 4-14	Standard menu dividers 63
	Figure 4-15	An inappropriate menu divider 63

Figure 4-16	A menu with text styles and an indicator	64
Figure 4-17	A checkmark to indicate a choice in a mutually exclusive group	65
Figure 4-18	A checkmark to indicate a choice in an accumulating attribute group	65
Figure 4-19	Dashes to indicate partial attributes in an accumulating attribute group	66
Figure 4-20	Several attributes in effect	67
Figure 4-21	The ellipsis character means more information is required	68
Figure 4-22	Don't use the ellipsis character with a command that doesn't require more information	69
Figure 4-23	The absence of the ellipsis character means no more information is required	70
Figure 4-24	The ellipsis character doesn't mean an alert box appears	71
Figure 4-25	The Application menu with a notification symbol	72
Figure 4-26	Don't use arbitrary symbols in menus	72
Figure 4-27	A Style menu with text styles	73
Figure 4-28	The effects of the two states of a Style menu item	74
Figure 4-29	A menu with nonstandard marks and extraneous text styles and a menu all in plain text style	75
Figure 4-30	A set of toggled menu items	76
Figure 4-31	A single toggled menu item whose name changes	76
Figure 4-32	An ambiguous toggled menu item	77
Figure 4-33	An incorrect use of a checkmark to indicate a state	78
Figure 4-34	A scrolling menu	78
Figure 4-35	The menu scrolling in the other direction	79
Figure 4-36	A hierarchical menu	79
Figure 4-37	Don't use submenus unnecessarily	80
Figure 4-38	A menu bar on a 9-inch screen with space for more menu titles	81
Figure 4-39	Examples of submenu titles	81
Figure 4-40	Avoid more than one level of submenus	82
Figure 4-41	A pop-up menu and its parts	83
Figure 4-42	Opening a pop-up menu	84
Figure 4-43	Pop-up menus versus radio buttons	85
Figure 4-44	Pop-up menus versus checkboxes	86
Figure 4-45	Don't use pop-up menus for commands	87
Figure 4-46	A standard pop-up menu	87
Figure 4-47	Using a pop-up menu	88
Figure 4-48	Correct and incorrect use of fonts in pop-up menus	89
Figure 4-49	Pop-up menu behavior	90
Figure 4-50	A hidden pop-up menu	91
Figure 4-51	A type-in pop-up menu	92
Figure 4-52	A type-in pop-up menu with user's choice added	92
Figure 4-53	A tools palette and a color palette	93
Figure 4-54	Using a tear-off menu	94
Figure 4-55	A tear-off menu on top of a document window	95
Figure 4-56	Palettes and feedback	96
Figure 4-57	A tool palette with the corresponding pointers	97
Figure 4-58	A tool palette in a window	97
Figure 4-59	An Apple menu	98
Figure 4-60	An About dialog box for an application	99

Figure 4-61	A File menu	99
Figure 4-62	The New command	100
Figure 4-63	The standard file dialog box for opening files	101
Figure 4-64	The save changes alert box	103
Figure 4-65	The correct location of the save changes alert box	104
Figure 4-66	The Save command	105
Figure 4-67	A sample alert box to use when a disk is full	105
Figure 4-68	The Save As command and dialog box	106
Figure 4-69	The Revert command	107
Figure 4-70	A Page Setup dialog box	108
Figure 4-71	A Print dialog box	109
Figure 4-72	A standard Edit menu for an application	110
Figure 4-73	Adding commands to the Edit menu	110
Figure 4-74	A sample Edit menu with Edition Manager commands	111
Figure 4-75	A sample hierarchical Edit menu with Edition Manager commands	111
Figure 4-76	The Clipboard	112
Figure 4-77	The Undo and Redo commands	114
Figure 4-78	The results of using the Paste command	116
Figure 4-79	The Create Publisher command and dialog box	118
Figure 4-80	The Subscribe To command and dialog box	118
Figure 4-81	The Publisher Options dialog box	119
Figure 4-82	The Subscriber Options dialog box	119
Figure 4-83	A Font menu	121
Figure 4-84	Don't combine the Font menu with other menus	122
Figure 4-85	A Size menu	122
Figure 4-86	A sample pull-down Size menu and font size dialog box	123
Figure 4-87	A Style menu	124
Figure 4-88	The Help menu	125
Figure 4-89	The Keyboard menu	126
Table 4-1	Apple-reserved keyboard equivalents for all systems	128
Table 4-2	Additional reserved keyboard equivalents for worldwide systems	128
Table 4-3	Common keyboard equivalents that are not reserved	129
Chapter 5	Windows	131
Figure 5-1	Examples of standard windows	133
Figure 5-2	Standard document window parts	134
Figure 5-3	Windows on a color screen	135
Figure 5-4	Standard window components in color	136
Figure 5-5	Colors that the user can choose for windows	137
Figure 5-6	A utility window	137
Figure 5-7	Make it clear where text will appear	138
Figure 5-8	The active window	139
Figure 5-9	Don't show a selection in an inactive window	141
Figure 5-10	Appropriate window titles for a series of unnamed windows	142
Figure 5-11	Examples of correct and incorrect window titles	143

Figure 5-12	Display order of document windows and modeless dialog boxes	144
Figure 5-13	Adding floating windows to the desktop	145
Figure 5-14	Adding a movable modal dialog box to the desktop	146
Figure 5-15	Window positions on a single screen	147
Figure 5-16	The standard window position on two sizes of screens	148
Figure 5-17	The standard window position on multiple screens	149
Figure 5-18	A window displayed across two screens	150
Figure 5-19	Standard position of an alert box	151
Figure 5-20	Alert box position in relation to the active document window	151
Figure 5-21	Standard alert box position with more than one screen	152
Figure 5-22	The close box	153
Figure 5-23	The save changes alert box	154
Figure 5-24	Moving a window	155
Figure 5-25	Multiple monitors and conceptual work space	156
Figure 5-26	A window growing larger	157
Figure 5-27	Relationship between a window and a document	158
Figure 5-28	The elements of a scroll bar	159
Figure 5-29	Using scroll arrows and the scroll box	159
Figure 5-30	Inactive scroll bars in active and inactive document windows	160
Figure 5-31	Background between the content and the window frame	161
Figure 5-32	Acceptable additions to the scroll bar region	162
Figure 5-33	Too many controls in the scroll bar	162
Figure 5-34	Scrolling by clicking a scroll arrow	163
Figure 5-35	Scrolling by clicking in the gray area	164
Figure 5-36	Scrolling by dragging the scroll box	165
Figure 5-37	Automatic scrolling	166
Figure 5-38	The zoom box	168
Figure 5-39	The standard state and the user state of a document	169
Figure 5-40	A split window	171
Figure 5-41	Split bar size	171
Figure 5-42	Independent and locked scrolling of window panes	172

Chapter 6

Dialog Boxes 175

Figure 6-1	Examples of dialog box types	177
Figure 6-2	A typical modeless dialog box	178
Figure 6-3	Two open modeless dialog boxes	179
Figure 6-4	The essential elements of a modeless dialog box	180
Figure 6-5	Incorrect absence of a close box in a modeless dialog box	181
Figure 6-6	Provide a place for the user to enter information in a modeless dialog box	184
Figure 6-7	A typical movable modal dialog box	185
Figure 6-8	The essential elements of a movable modal dialog box	186
Figure 6-9	Close box used incorrectly in a movable modal dialog box	187
Figure 6-10	A Finder movable modal dialog box	187
Figure 6-11	Menu bar access while a movable modal dialog box is open	188
Figure 6-12	An example of a modal dialog box	189
Figure 6-13	A status dialog box	190

Figure 6-14	The essential elements of a modal dialog box	190
Figure 6-15	Access to the Edit menu when displaying a modal dialog box	191
Figure 6-16	Second modal dialog box on top of first one	193
Figure 6-17	The essential elements of an alert box	194
Figure 6-18	An example of a note alert box	195
Figure 6-19	An example of a caution alert box	195
Figure 6-20	An example of a stop alert box	196
Figure 6-21	Recommended spacing of buttons and text in dialog and alert boxes	197
Figure 6-22	An active scrolling list	198
Figure 6-23	A well-written dialog box message	199
Figure 6-24	The standard file dialog box for opening files	200
Figure 6-25	The save changes alert box	201

Chapter 7

Controls 203

Figure 7-1	Buttons in a dialog box	205
Figure 7-2	A highlighted button	205
Figure 7-3	A dialog box with OK and Cancel buttons	207
Figure 7-4	A dialog box with a Done button instead of an OK button	208
Figure 7-5	A progress indicator that uses a Stop button	209
Figure 7-6	A confirmation alert box with appropriately named button	209
Figure 7-7	Sets of radio buttons	210
Figure 7-8	Radio buttons for selecting the alignment of text	211
Figure 7-9	The General Controls panel	211
Figure 7-10	A set of checkboxes	212
Figure 7-11	A single checkbox in a dialog box	212
Figure 7-12	The Find dialog box	213
Figure 7-13	An example of a slider	214
Figure 7-14	A slider with direction information	215
Figure 7-15	Incorrect use of a scroll bar and correct use of a slider	215
Figure 7-16	Little arrows control	216
Figure 7-17	Content-dependent increment	217
Figure 7-18	Outline triangle control	218
Figure 7-19	A text entry field	219
Figure 7-20	A scrolling list	220

Chapter 8

Icons 223

Figure 8-1	Common icons	224
Figure 8-2	Examples of common traffic symbols	225
Figure 8-3	Examples of commonly-used international symbols	225
Figure 8-4	Symbols are easier to understand than keyboard commands	226
Figure 8-5	Grouping icons on the desktop	226
Figure 8-6	A confusing image	227
Figure 8-7	Context clarifies the image	227
Figure 8-8	Icons with label text	228
Figure 8-9	A logical and an illogical metaphor	229

Figure 8-10	Localized mailbox icons	230
Figure 8-11	Avoid text in icons	231
Figure 8-12	Certain shapes don't work well	231
Figure 8-13	A consistent light source	232
Figure 8-14	Inconsistent light sources	232
Figure 8-15	Design the large icon first and base the small icon design on it	233
Figure 8-16	Consistent use of icon elements	234
Figure 8-17	An icon family	234
Figure 8-18	Different sizes of icons	235
Figure 8-19	A well-designed icon and its selected version	238
Figure 8-20	A poorly designed icon and its selected version	238
Figure 8-21	Icons with a black outline	239
Figure 8-22	Icons without a black outline	239
Figure 8-23	Standard 256-color palette with icon colors marked	240
Figure 8-24	An example of dithered color in an icon	241
Figure 8-25	Color icons and their selected states	242
Figure 8-26	Color icons and their color-labeled states	243
Figure 8-27	Correct anti-aliasing	243
Figure 8-28	Consistently designed small icons	244
Figure 8-29	Inconsistently designed small icons	245
Figure 8-30	Default application icons	246
Figure 8-31	Custom application icons	246
Figure 8-32	Examples of bad application icons	247
Figure 8-33	Default document icons	247
Figure 8-34	Application icon and document icon with the same graphic element	247
Figure 8-35	Acceptable and unacceptable custom document icons	248
Figure 8-36	Document icons with standard symbols	248
Figure 8-37	Default stationery pad icons	249
Figure 8-38	Default query document icons	249
Figure 8-39	Default edition icons	250
Figure 8-40	Preferences file icons	250
Figure 8-41	Default extension icons	251
Figure 8-42	Examples of Chooser icons	251
Figure 8-43	Icons for the Color control panel	251
Figure 8-44	Font icons	252
Figure 8-45	A sound icon	252
Figure 8-46	The default keyboard layout and input method icons	253
Figure 8-47	Examples of keyboard icons	253
Figure 8-48	Examples of modification indicators on keyboard icons	254
Figure 8-49	Enlarged keyboard icons with correct color substitutions	255
Table 8-1	Icon display on monitors of different bit depths	235
Table 8-2	Pattern substitutions for colors in keyboard icons	254

Chapter 9**Color 257**

- Figure 9-1** A colorized window 259
Figure 9-2 A colorized movable modal dialog box 259
Figure 9-3 Color palette and custom color mixing tool 262
Figure 9-4 Design for black-and-white monitors first 263
Figure 9-5 Don't mimic color effects in black-and-white designs 263
Figure 9-6 A limited palette of colors 264

Chapter 10**Behaviors 267**

- Figure 10-1** Different pointing devices 268
Figure 10-2 The insertion point and the pointer 269
Figure 10-3 A status indicator 271
Figure 10-4 Clicking a button 272
Figure 10-5 Double-clicking to select a word 272
Figure 10-6 Pressing a scroll arrow 273
Figure 10-7 Dragging to move an object 274
Figure 10-8 Using the Tab key to cycle through fields 276
Figure 10-9 Using the Return key to move the insertion point 277
Figure 10-10 A sample confirmation dialog box for the Escape key 278
Figure 10-11 Using Option-drag to make a copy of an object 279
Figure 10-12 Arrow keys 281
Figure 10-13 The function keys 284
Figure 10-14 Three ways of selecting information 287
Figure 10-15 Selection techniques 288
Figure 10-16 Expanding and shrinking a text selection 290
Figure 10-17 Extending text selections using the addition and fixed-point methods 290
Figure 10-18 Discontinuous selection within an array 292
Figure 10-19 Text selections 293
Figure 10-20 Selecting with Shift and arrow keys 296
Figure 10-21 Selecting with Option-Shift and arrow keys 296
Figure 10-22 Selection in an object-based graphics document 297
Figure 10-23 Selection in a bitmap-based graphics document 297
Figure 10-24 Field selection in an array 298
Figure 10-25 Column selection in an array 298
Figure 10-26 Range selection in an array 299
Figure 10-27 Discontinuous selection in an array 299
Figure 10-28 Intelligent cut and paste 302
- Table 10-1** Pointers 270
Table 10-2 How modifier keys change the movement of the insertion point with the arrow keys 283

Chapter 11

Language 305

Figure 11-1	Proper capitalization of screen elements	309
Figure 11-2	Clear button names	310
Figure 11-3	A poorly written alert box message	311
Figure 11-4	An improved alert box message	311
Figure 11-5	A well-written alert box message	311
Figure 11-6	Correct absence of a colon to introduce a list of options	312
Figure 11-7	Correct use of a colon	312
Figure 11-8	A help balloon	316
Figure 11-9	Help balloon for a button	319
Figure 11-10	Help balloon for a menu title	320
Figure 11-11	Help balloon for a menu item	320
Figure 11-12	Help balloon for a selected radio button	321
Figure 11-13	Help balloon for a checkbox	322
Figure 11-14	Help balloon for a group of radio buttons	323
Figure 11-15	Help balloon for a tool palette	323
Figure 11-16	Help balloons for an application icon and a document icon	324
Figure 11-17	Help balloon for a text entry box	325
Table 11-1	Translation chart for user documentation	307
Table 11-2	Categories of questions for help systems	315

About This Book

Macintosh Human Interface Guidelines describes the way to create products that optimize the interaction between people and Macintosh computers. It explains the whys and hows of the Macintosh interface in general terms and specific details.

Macintosh Human Interface Guidelines helps you link the philosophy behind the Macintosh interface to the actual implementation of interface elements. Examples from a wide range of Macintosh products show good human interface design, including individual and iterative examples. These examples are accompanied by descriptions and discussions of why to follow the guidelines. This book also contains examples of how *not* to do human interface design; they are marked as such and appear with a discussion that points out what's inappropriate and how to correct it.

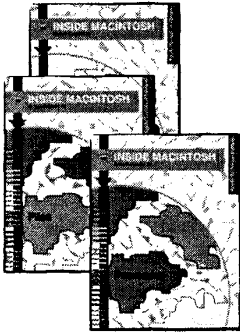
Who Should Read This Book

This book is written for people who design and develop products for use with Macintosh computers. If you are a designer, a human interface professional, or an engineer, this book contains information you can use to design and create products that fit the Macintosh model. It provides background information that can help you plan and make decisions about your product design.

Even if you don't design and develop products for the Macintosh, reading this book will help you to understand the Macintosh interface. This book is appropriate for managers and planners who are thinking about developing Macintosh products, as well as people who are interested in human interface design in general.

This book is written with the assumption that you are familiar with the concepts and terminology used with Macintosh computers and that you have used a Macintosh computer and some Macintosh applications.

What's New in Macintosh Human Interface From Apple



In previous years, the human interface guidelines were published in *Inside Macintosh*, in *Human Interface Guidelines: The Apple Desktop Interface*, and in *Human Interface Notes*. This book contains all the current human interface guidelines pertaining to all types of products that work with Macintosh computers.

In the current *Inside Macintosh*, you'll find human interface information and guidelines included in the individual chapters about each topic, such as the Edition Manager or the Dialog Manager. These chapters also have technical implementation information about the Macintosh Operating System. *Inside Macintosh* doesn't contain the philosophy of the Macintosh interface or the rationale about how to use the interface elements properly, so to create your product, you'll need *Inside Macintosh* and this book.

About *Making It Macintosh*



Making It Macintosh: Macintosh Human Interface Guidelines Companion is a CD-ROM-based accompaniment to *Macintosh Human Interface Guidelines*. *Making It Macintosh* contains animated examples of common problems developers encounter when they implement the Macintosh interface, presents solutions to these problems, and discusses how you can approach similar problems. You can use *Making It Macintosh* and *Macintosh Human Interface Guidelines* together, or you can consult each separately. For information about ordering *Making It Macintosh*, see the back of this book.

What's in This Book

This book contains three major sections. You can read it from start to finish or use it as a reference in which to look up specific pieces of information that you need to know. The paragraphs that follow describe the type of information you'll find in each part of the book.

The Basic Philosophy

The first part of this book presents the key design principles and considerations for developers to keep in mind when creating a product that works with Macintosh computers. It suggests ways to incorporate human interface into your product design and decision-making processes.

P R E F A C E

It also describes how to involve users in your product design process to ensure that you've built a product that serves your users well and conforms to the guidelines.

The Interface Elements

The second part of this book defines various parts of the Macintosh interface. It presents examples of the right and wrong ways to use interface elements and behaviors and contains specific implementation information you can use while you're creating a product. This part of the book also shows how to combine interface elements with behaviors, aesthetics, and language to create a superior product.

Appendixes

The appendixes provide additional information about the topics discussed in this book. Appendix A describes resources such as professional societies and conferences from which you can get additional information. Appendix B is a bibliography that presents major works on topics discussed in the book. Refer to this appendix when you want to find where to get more extensive information or training on a topic such as color or menus. Appendix C provides a checklist for you to use when evaluating your product to make sure it meets the intent and purpose of the Macintosh human interface guidelines. The glossary, which follows the appendixes, provides definitions for terms used in the book. At the end of the book is an index, which will help you locate information about specific topics.

Visual Cues Used in This Book



In this book you'll find visual cues to certain types of information.



- This symbol indicates an example of the correct way to use an interface element.
- This symbol indicates an example that is OK or is an improvement to a poor example. It would still be useful to consider making improvements to this type of example.
- This symbol indicates an example of the wrong way to use an interface element. It specifically calls out common mistakes.
- **Boldfaced text** indicates that a new term is being defined and that a definition of the word appears in the glossary.

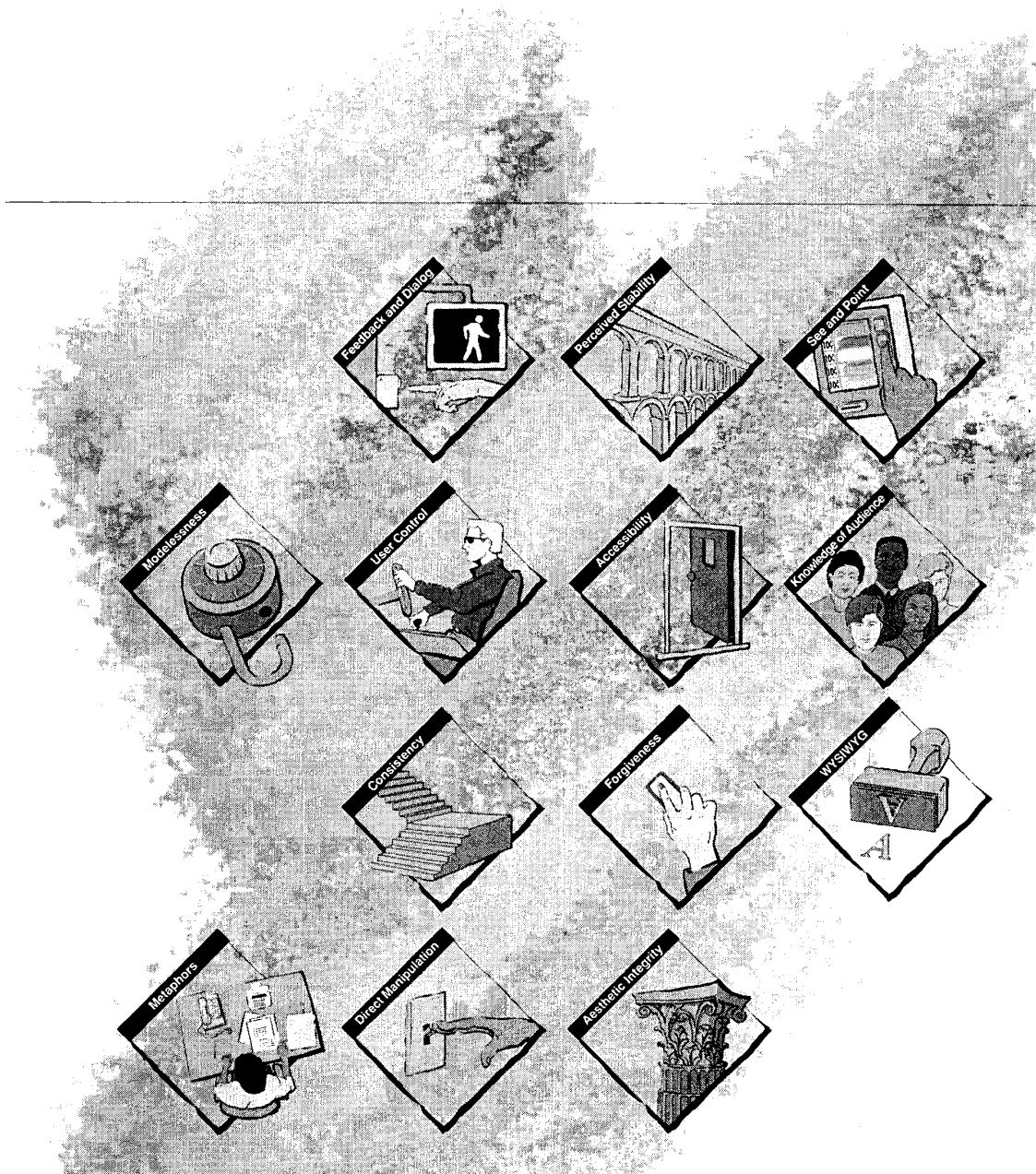
Fundamentals

This part of *Macintosh Human Interface Guidelines* presents the philosophy and psychology behind the Macintosh interface. Read this part to learn about the design principles and considerations that developers can use to create an excellent human interface. You can find out how to incorporate good human interface into your design and decision-making processes and how to involve users throughout the design process. You can also read about how to work with and go beyond the guidelines while maintaining their spirit and intent.

This part contains the following chapters:

- Human Interface Principles
- General Design Considerations
- Human Interface Design and the Development Process

Human Interface Principles



Human Interface Principles

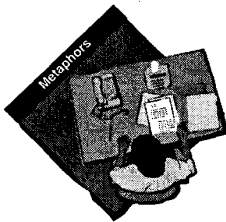
At Apple Computer, products are designed with a number of basic principles of human-computer interaction in mind. This chapter discusses the human interface design principles that describe key considerations for the design decisions you make for your product.

Having technical knowledge of the Macintosh user interface is a key factor in product design, but understanding the theories behind the user interface can help you create an excellent product. This chapter provides a theoretical base for the wealth of practical information on implementing the Macintosh interface elements presented in Part Two.

The Human Interface Design Principles

This section presents a set of principles useful for designing products for Macintosh computers. The reason for defining a set of design principles is to help you build a product that meets the standards of the Macintosh computer. Also, these principles can help clarify what you can do in order to design a product based on what is known about people and how they operate in the world.

You'll undoubtedly find out that you can't design in accordance with all of the principles all of the time. In that type of situation, you'll have to make a decision based on which principle or set of principles is most important in the context of the task you're solving.



Metaphors

You can take advantage of people's knowledge of the world around them by using metaphors to convey concepts and features of your application. Use metaphors involving concrete, familiar ideas and make the metaphors plain, so that users have a set of expectations to apply to computer environments. For example, people often use file folders to store paper documents in their offices. Therefore, it makes sense to people to store computer documents in computer-generated folders that look like file folders. People can organize their hard disks in a way that's analogous to the way they organize their file cabinets.

The desktop is the primary metaphor for the Macintosh interface. It appears to be a surface on which people can keep tools and documents. Several other metaphors are integrated into the desktop metaphor. It makes sense in the context of a desktop environment to include folders and a trash can (even though most trash cans don't sit on the desktop). Menus are an extension of the desktop metaphor. People can connect the idea of making choices from a computer menu with making choices from a restaurant menu. Although people don't keep restaurant menus on the edge of their desks, using the term *menu* in the computer environment reinforces the idea that people can use computer menus to make choices.

Metaphors in the computer interface suggest a use for something, but that use doesn't define or limit the implementation of the metaphor. For example, a paper file folder has a limited storage capacity, but a folder on the Macintosh doesn't have to be constrained by the same limitations. Computer folders can hold a limitless number of files (up to the storage capacity of the hardware), and this is an advantage that the computer can offer. Try to strike a balance between the metaphor's suggested use and the ability of the computer to support and extend the metaphor.



Direct Manipulation

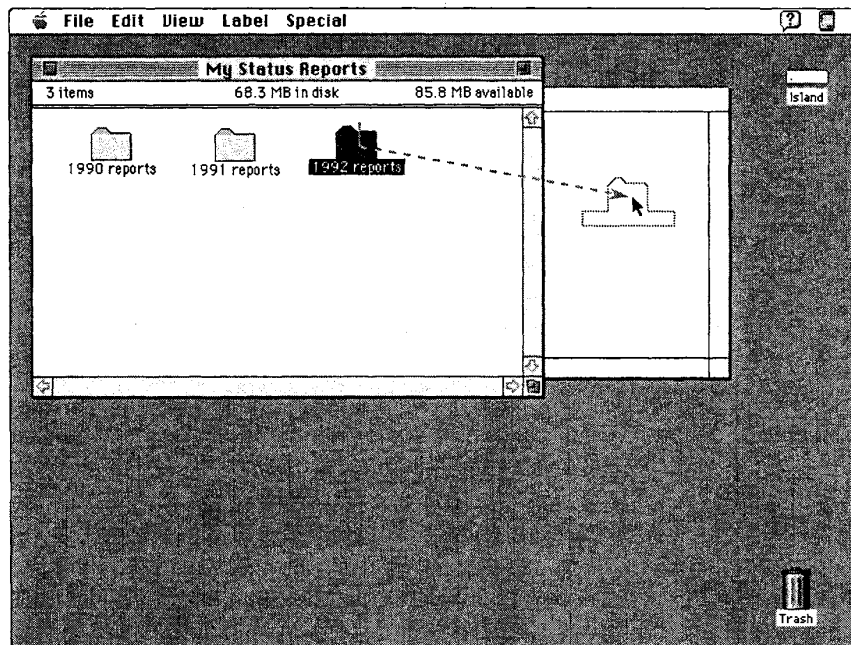
Direct manipulation allows people to feel that they are directly controlling the objects represented by the computer. According to the principle of direct manipulation, an object on the screen remains visible while a user performs physical actions on the object. When the user performs operations on the object, the impact of those operations on the object is immediately visible. For example, a user can move a file by dragging an icon that represents it from one location to another or can position a cursor in a text field by directly clicking the location where the cursor should be placed.

CHAPTER 1

Human Interface Principles

Figure 1-1 shows a folder icon being dragged across the desktop as an example of direct manipulation on the computer.

Figure 1-1 Direct manipulation



In addition to expecting physical results from their actions, users want their tools to provide feedback. For example, when a drawing tool is moved, a line appears in the document on which the user is working. Users want to see what actions are available at any given moment. If grave consequences might follow from any of those actions, they want to know about those consequences—before any damage is done and while they can still change their minds. They want clues that tell them that a particular command is being carried out, or, if it cannot be carried out, they want to know why not and what they can do instead. Users also want topics of interest to be highlighted.

Animation, when used sparingly, is one of the best ways to show a user that a requested action is being carried out. For example, animated pointers reassure the user, during a lengthy process such as saving a large document to disk, that the computer is completing the task without any problems.



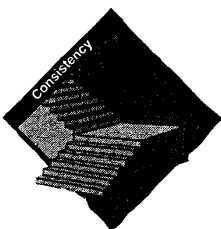
See-and-Point

On the desktop, users perform actions by choosing from alternatives presented on the screen. Users interact directly with the screen, selecting objects and performing activities by using a pointing device, typically a mouse, to point at elements on the desktop.

The Macintosh desktop works according to two fundamental paradigms. Both paradigms share two basic assumptions: that users can see on the screen what they're doing and that users can point at what they see. The paradigms are based on a general form of user action: noun-then-verb.

In one paradigm, the user selects an object of interest (the noun) and then chooses the actions to be performed on the object (the verb). All actions available for the selected object are listed in the menus, so users who are unsure of what to do next can refresh their memory by scanning through the menus. At any time, users can choose any available action without having to remember any particular command or name. For example, a user clicks a document icon (the noun) and then prints (the verb) the document by choosing Print from the File menu.

In the second paradigm, the user drags an object (the noun) onto some other object that has an action (the verb) associated with it. On the desktop, for example, the user can drag icons to the Trash, to folders, or to disks. The user doesn't choose an action from the menus, but it's clear what happens to one object when it's placed on another object. For example, dragging a document icon to the Trash means that the user wants to discard that document. For this metaphor to work, the user must recognize what an object such as the Trash is for, so it is especially important that objects look like what they do in the real world. If the document icon didn't look like a piece of paper with text and the Trash didn't look like the place to discard something, the interface would be more difficult to use.



Consistency

Consistency in the interface allows people to transfer their knowledge and skills from one application to any other. Use the standard elements of the Macintosh interface to ensure consistency within your application and to benefit from consistency across applications.

Effective applications are consistent in a number of different ways. Consistency in the visual interface helps people learn and then easily recognize the graphic language of the interface—for example, once users know what a checkbox looks like, they don't have to learn another symbol for making choices. Consistency in the behavior of the interface means that people have to learn how to do things such as clicking and pointing only once; then they can explore new applications or new types of features using

Human Interface Principles

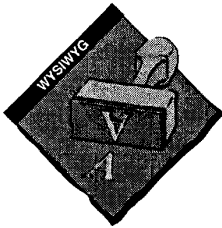
skills that they already have. In general, consistency benefits the typical user, who usually divides working time among several applications, and it benefits software developers because their users can build on prior experiences with elements in other applications when learning how to use a new application.

The following are some questions you can ask yourself when thinking about consistency in your product.

Is your product consistent

- within itself?
- with earlier versions of your product?
- with Macintosh interface standards?
- in its use of metaphors?
- with people's expectations?

Note that the most difficult kind of consistency to achieve is matching people's expectations. Because you often face a wide audience and a range of expertise, it's difficult to meet the expectations of everyone. You can address this problem by carefully weighing the consistency issues in the context of your target audience and their needs.



WYSIWYG (What You See Is What You Get)

Don't hide features in your application by using abstract commands. People should be able to see what they need when they need it. For example, menus present lists of commands so that people can see their choices instead of having to remember and type command names.

People should be able to find all the available features in your application. If you find a need to initially "hide" features, do it in a way that gives people information about where they can find more choices. A stepped interface, by revealing relevant information to users in steps, shows the choice most users want most of the time while providing a way for the user to get more choices. For information on stepped interfaces, see the guidelines in the section "Using Progressive Disclosure" on page 35 in Chapter 3, "Human Interface Design and the Development Process."

Make sure that there is no significant difference between what the user sees on the screen and what the user receives after printing. Let the user be in charge of both the content and the format (spatial layout as well as font choices) of the document. When the user makes changes to the document, quickly and directly display the results; the user shouldn't have to wait for a printout or make mental calculations of how the document shown on the screen will look when it appears on the printed page.



User Control

Allow the user, not the computer, to initiate and control actions. People learn best when they're actively engaged. Too often, however, the computer acts and the user merely reacts within a limited set of options. In other instances, the computer "takes care" of the user, offering only those alternatives that are judged "good" for the user or that "protect" the user from having to make detailed decisions. This approach mistakenly puts the computer, not the user, in control.

The key is to create a balance between providing users with the capabilities they need to get their work done and preventing them from destroying data. For situations in which a user may destroy data accidentally, you can help the user by providing warnings, usually in the form of an alert box, to notify users of a potentially undesirable situation and still allow them to proceed, if they confirm that this is what they want. This approach "protects" users but allows them to remain in control.



Feedback and Dialog

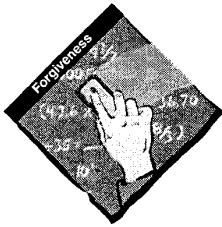
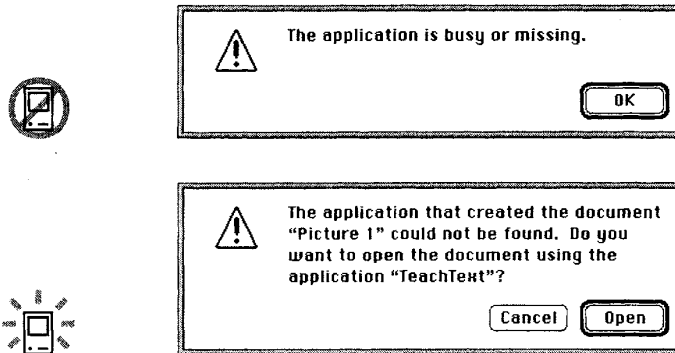
Keep users informed about what's happening with your product. Provide feedback as they do tasks and make that feedback as immediate as possible. When a user initiates an action, provide some indicator, visual or auditory (or both), that your application has received the user's input and is operating on it. Provide as much information as possible about how long operations take. When your application can't respond to user input because it's processing a different task, inform the user of what to expect and describe any delays, why they occur, and how long they'll take. Also, tell the user how to get out of the current situation whenever possible.

Provide direct, simple feedback that people can understand. Most people would not know what to do if they saw this message "The computer unexpectedly crashed. ID = 13." It would be very helpful if the message spelled out exactly which situation caused the error—for example, not enough memory was available for the computer to complete the task—so that the user could understand how to avoid the situation in the future.

Human Interface Principles

Figure 1-2 shows an example of a message that doesn't do anything to help the user and an example of a message that provides useful and helpful information to the user.

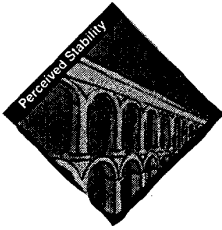
Figure 1-2 An example of a bad message and an example of a helpful message



Forgiveness

You can encourage people to explore your application by building in forgiveness. Forgiveness means that actions on the computer are generally reversible. People need to feel that they can try things without damaging the system; create safety nets for people so that they feel comfortable learning and using your product.

- Always warn people before they initiate a task that will cause irretrievable data loss. Alert boxes are a good way to warn users of this kind of situation. Note, however, that when options are presented clearly and feedback is appropriate and timely, learning how to use a program should be relatively error-free. This means that frequent alert boxes are a good indication that something is wrong with the program design.

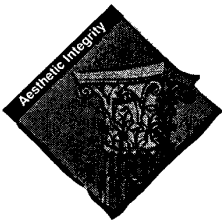


Perceived Stability

Computers often introduce a new level of complexity for people. If people are to cope with this complexity, they need some stable reference points. The Macintosh interface is designed to provide a computer environment that is understandable, familiar, and predictable.

To give users a visual sense of stability, the Macintosh interface provides the desktop, a two-dimensional space on which objects are placed. It also defines a number of consistent graphics elements (menu bar, window border, and so on) to maintain the illusion of stability. Note that it is the *perception* of stability that you want to preserve, not stability in any strict physical sense.

To give users a conceptual sense of stability, the interface provides a clear, finite set of objects and a clear, finite set of actions to perform on those objects. Even when particular actions are unavailable, they are not eliminated from a display but are merely dimmed.



Aesthetic Integrity

Aesthetic integrity means that information is well organized and consistent with principles of visual design. This means that things look good on the screen and the display technology is of high quality. Since people spend a lot of their time working while looking at the computer screen, design your products to be pleasant to look at on the screen for a long time. You may want to consider investing some of your resources in a graphic designer; the skills a graphic designer can bring to your product design are well worth the expense.

Keep the graphics of the display simple. The number of elements and their behaviors should be limited to enhance the usability of the interface.

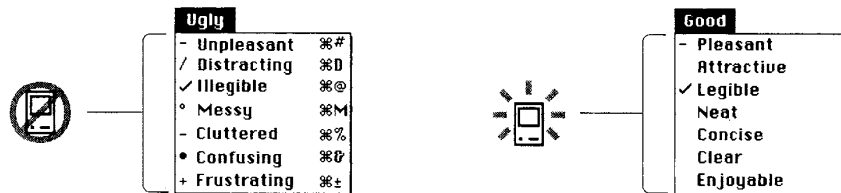
Graphics—icons, windows, dialog boxes, and so on—are the basis of effective human-computer interaction and must be designed with that in mind. Don't clutter the screen with too many windows, overload the user with complex icons, or put dozens of buttons in dialog boxes.

Make sure to follow the graphic language of the interface and don't change the meaning of standard items. For example, if you sometimes use checkboxes for multiple choices and other times for exclusive choices, you dilute the meaning of the element.

Don't use arbitrary graphic images to represent concepts. When you add nonstandard symbols to menus, dialog boxes, or other elements, the meaning may be clear to you, but to other people the symbols may appear as something different and distracting. If you need symbols other than standard ones, use graphic images that convey meaning through representation, analogy, or metaphor. For more information on designing additional appropriate symbols, see the section "Extending the Interface" in Chapter 3, "Human Interface Design and the Development Process," beginning on page 38.

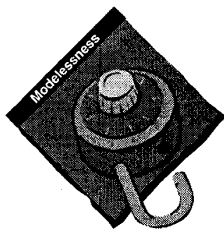
Figure 1-3 shows an example of how confusing arbitrary symbols can be and how much clearer a simple menu with standard symbols can be.

Figure 1-3 Don't use arbitrary graphic elements



In general, match the graphic element with users' expectations of its behavior. Push buttons appear as though they push in rather than slide sideways. Indicators in sliders slide along to change values. These behaviors map to people's expectations of how these elements behave.

Give users some control over the look of their computer environments. This allows them to display their own style and individuality. It also reduces the burden on the designer of trying to create an interface that appeals to every user. When a user sets up his or her computer environment in a certain layout, it should stay that way until the user changes it.



Modelessness

For the most part, try to create modeless features that allow people to do whatever they want when they want to in your application. Avoid using modes in your application because a mode typically restricts the operations that the user can perform while it is in effect. It locks the user into one operation and doesn't allow the user to work on anything else until that operation is completed. In contrast, modelessness allows the user to perform more than one operation at a time and thus gives the user more control over what he or she can do on the computer and in an application. As much as possible, you want to preserve the user's ability to be in control of the task and the order of operations.

This is not to say that you should never use modes in applications. Sometimes using a mode is the best way out of a particular problem. Most acceptable modes fall into one of the following categories:

- Long-term modes, such as doing word processing as opposed to graphics editing. In this sense, each application is a mode.

- Short-term “spring-loaded” modes, in which the user must constantly do something to maintain the mode. Examples are holding down the mouse button to scroll text or holding down the Shift key to extend a text selection.
- Alert modes, in which the user must rectify an unusual situation before proceeding. Keep these modes to a minimum.

Other modes are acceptable if they do one of the following:

- They emulate a familiar real-life situation that is itself modal. For example, choosing different tools in a graphics application resembles the real-life choice of physical drawing tools.
- They change only the attributes of something, not its behavior. The boldface and underline modes of text entry are examples.
- They block most other normal operation of the system to emphasize the modality, as in error conditions incurable through the software (for example, a dialog box that disables all menu items except Close).

If an application uses modes, there must be a clear visual indicator of the current mode, and the indicator should be near the object most affected by the mode. A good example is the changing pointer in many Macintosh graphics applications; depending on the function (“mode”) the user has selected, the pointer looks like a pencil, a paintbrush, a spray can, or an eraser. It should also be very easy for users to get into or out of the mode (such as by clicking a different palette symbol).

Additional Issues to Consider

This section discusses several other issues that are helpful to think about when you design your product.



Knowledge of Your Audience

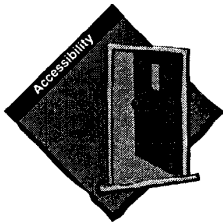
Identifying and understanding your target audience are among the most important first steps when you start designing your product. To create a product that people can and will use, study the people who make up your target audience.

It's useful to create scenarios that describe a typical day in the life of a person you think uses the type of product you're designing. Think about the different work spaces, tools, and constraints and limitations that people deal with. You can also visit actual work places and study how people do their jobs.

Analyze the steps necessary to complete each task you anticipate people wanting to accomplish. Then design your product to facilitate those tasks, using a step-by-step approach by thinking of how a person might get from one place to the next in a logical fashion.

Human Interface Principles

Involve users throughout the design process and observe them working in their environment. Use people who fit your audience description to test your prototypes and development products. Listen to their feedback and try to address their needs in your product. Develop your product with people and their capabilities, not computers and their capabilities, in mind. For more information, see the section “Involving Users in the Design Process” in Chapter 3, “Human Interface Design and the Development Process,” beginning on page 41.



Accessibility

The computer should be accessible to everyone who chooses to use it. There are likely to be members of your target audience who are different from the “average” user that you envision. Users will undoubtedly vary in their ages, styles, and abilities. They may also have physical or cognitive limitations, linguistic differences, or other differences you need to consider. Identify how the individuals in your target audience differ and what special needs they may have.

Make it easy for users to interact with your product using different input devices and output devices. If you develop specialized hardware and software for people with physical limitations, work with application developers so that your products are supported by their software.

Make your application accessible to people around the world by including support for worldwide capabilities in your designs from the beginning of your development process. Take stock of the cultural and linguistic needs and expectations of your target audiences. For more information, see the section “Worldwide Compatibility” beginning on page 16 in Chapter 2, “General Design Considerations.”

More information about universal access appears throughout the book where appropriate. For specific information, see the section “Universal Access” beginning on page 24 in Chapter 2, “General Design Considerations.”