

Exhibit 1030



Macintosh Human Interface Guidelines

by Apple Computer, Inc.

Apple Technical Library



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

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

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

Human Interface Design and the Development Process

This chapter discusses aspects of the product development process, including ways you can incorporate human interface design into that process. First, this chapter presents several issues to consider when deciding on the features you want to include in your product. Second, this chapter provides suggestions for how to make your product simple and easy to use; too often, users are confused by unnecessary complexity in an application. Next, this chapter describes how to extend the interface when your application requires functionality that isn't covered by existing user interface elements. Finally, this chapter discusses the benefits of involving users throughout the entire phase of your product development process.

Design Decisions

For design decisions regarding features in your application, you need to weigh the costs, which are not all financial, against the potential benefits. Every time you add a feature to your application, think about the following factors:

- Your application gets larger.
- Your application gets slower.
- Your application's human interface gets more complex.
- You spend time in development rather than refinement of existing features.
- Your application becomes more difficult to document.
- You increase the number of possible user errors.
- Every new feature can have an impact on an existing feature.

This section presents several additional factors you'll want to take into consideration when adding features to your product.

Features Inspired by Market Pressures

Remember to think about your users and keep their interests foremost when making your design decisions. Market pressures can sometimes cause developers to try to implement features that they feel are necessary, even when they don't have the resources to develop those features fully. Because a good review in the press can make an application successful, developers are sensitive to meeting the expectations of reviewers as well as their target audiences. Many times reviews include information on whether or not applications have certain features and the right number of new features. The pressure of competition is intense, and developers may often feel that they must make decisions based on market pressures.



Feature Cascade

When deciding whether or not to add features to your product, think about whether the benefits to users of additional capabilities outweigh the additional development efforts, growth in size, and reduction in running speed that the features would cost. If you are developing a simple application, it's very tempting to include additional features that users claim they want. It takes a lot of restraint to stick to the original intent of the application. Watch out for feature cascade, because it can often reduce the overall effectiveness of and add unwanted complexity to your application.

The 80 Percent Solution

During the design process, you may discover problems with your product design. You can use the 80 percent solution to help determine how to solve those problems. The 80 percent solution means that your design meets the needs of at least 80 percent of your users. If you try to design for the 20 percent of your target audience who are power users, your design will not be usable by the majority of your users. Even though those 20 percent are likely to have good ideas and probably think a lot like you, the majority of people may not be like the 20 percent who are elite users of your product. Involving a broad range of users in your design process will help you find the 80 percent solution.

Managing Complexity

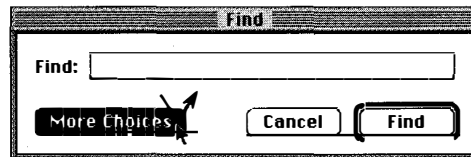
The best approach to developing software that is easy to use is to keep the design as simple as possible. (The general design guideline that simple design is good design applies directly to the discipline of human-computer interactions.) The challenge that users desire is to solve their problems using the tools that you design to facilitate their work. The more you can do to simplify the interface and your product for your users, the more likely it is that you will build a product that meets their needs and expectations.

Using Progressive Disclosure

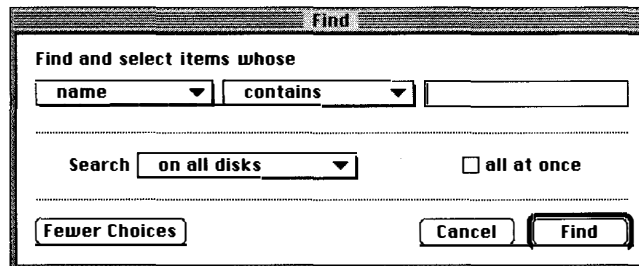
Progressive disclosure is one way to reduce the complexity of your designs. It allows you to present the most common choices to users while initially hiding more complex choices or additional information. Progressive disclosure helps you develop your interface so that it is easy for novice users to learn and includes the features and power that advanced users desire.

For dialog boxes, you can implement progressive disclosure and thus reduce the complexity of your design by presenting only the most common options in the dialog box that appears initially on the screen. There are a couple of techniques you can implement to allow users to see more options. The most standard method of letting users see more choices is including a button named More Choices in the lower-left corner of the dialog box. When the user clicks the button, the dialog box expands to display more information and the button name changes to Fewer Choices. This method is very clear and predictable. People know how to use buttons, and the labels let people know right away that they have access to more information if they don't see what they're looking for. Figure 3-1 shows an example of a dialog box that implements this method of progressive disclosure.

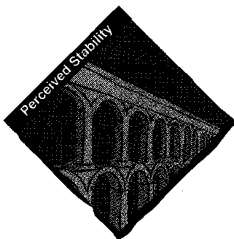
Figure 3-1 An expanding dialog box



1.



2.



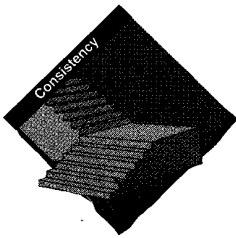
If you include a More Choices button in a dialog box, it's best to extend the bottom of the window to accommodate the additional information. You could move a side of the window without causing too much change in the perceived stability of the window. However, avoid expanding the window symmetrically in all directions, because this behavior destroys the user's initial sense of the window being one object with some of it not visible. In general, it's best to keep all controls visible at all times. Also, if the larger state of the dialog box won't fit on the screen when the More Choices button is selected, move the dialog box the minimal amount necessary to make it fit on the screen. When the user clicks the Fewer Choices button, keep the dialog box in its new position; *don't* move it back to its original position on the screen.

In order to reduce the complexity of your application, make decisions about which features to implement as preferences based on what your users really need. The key is to implement as preferences only those features that your users find useful. In other words, avoid creating one large dialog box with all the preferences you can think of. Instead, eliminate the settings that are special cases of a behavior or an attribute and build in flexible features as a part of your application.

A preference should be a setting that the user changes *infrequently*. If you provide choices to users that they will change many times in a work session, you should implement those choices in a menu or other interface element to which the user has easy, modeless access. By choosing the right way to implement a feature, you can give users the flexibility to choose, in their own language, their preferred method of working.

Extending the Interface

This section describes how to extend the Macintosh user interface when your application needs an element that doesn't already exist. When a need arises that can't be met by the standard elements, you can extend the user interface by creating combinations of standard elements or new elements. This section contains information on how to determine when it's appropriate to go beyond the guidelines, how to use the existing interface elements to build new elements, and pitfalls to avoid when you design additional interface elements.



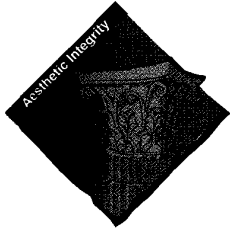
When to Go Beyond the Guidelines

People rely on the standard Macintosh user interface for consistency. Don't copy other platforms' user interface elements or behaviors in the Macintosh because they may confuse users who aren't familiar with them.

There are times when the standard user interface doesn't cover the needs of your application. This is true in the following situations:

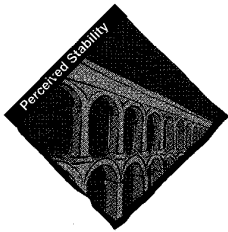
- You are creating a new feature for which no element or behavior exists. In this case, you can extend the Macintosh user interface in a prescribed way.
- An existing element does almost everything you need it to, but a little modification that improves its function makes the difference to your application.

The sections that follow present the guidelines that describe how to extend the Macintosh user interface guidelines.



Build on the Existing Interface

When you need to extend the user interface, the best place to begin is with the already defined visual and behavioral language. Look carefully at the elements that are defined in this book. Think about what the appearance means to people (the look) and how they expect the element to behave (the feel). Visual cues, like the drop shadow and the arrow on a pop-up menu, are triggers for people. These cues help people recognize elements that they can use. People also learn to associate certain behaviors with specific elements. For example, people recognize push buttons by their rounded rectangle shape. They look for a label that identifies the action the button causes. This particular appearance distinguishes a push button from other types of elements. When people click a button, they expect the button to be highlighted to indicate that the action takes effect. People may also expect that clicking a button has additional behaviors related to it, including dismissing a dialog box or changing the content area of the active document. Mixing visual cues is confusing to users. For example, adding a drop shadow to a push button makes the push button look like a pop-up menu.



Don't Assign New Behaviors to Existing Objects

The previous section describes how to use elements from the existing user interface. When you do use existing interface building blocks, use them in the standard way. Make sure you do *not* change the behavior for standard elements. When you need a new behavior, design a new element for it. If elements behave differently in different situations, the interface becomes unpredictable and harder to figure out.

Consider an example of a palette that includes a subpalette. You might think of using the ellipsis character to indicate that the palette will display additional choices. However, using this symbol to indicate a subpalette gives it a meaning other than the standard meaning. In menus, the ellipsis character means that the user must provide more information before a command will operate. It doesn't mean that more information appears when the user chooses the item. This incorrect use of the ellipsis character is shown in Figure 3-3.