

# HyperCard

.....  
*Installation and new features*

 Apple Computer, Inc.

© 1998 Apple Computer, Inc. All rights reserved.

Under the copyright laws, this manual may not be copied, in whole or in part, without the written consent of Apple. Your rights to the software are governed by the accompanying software license agreement.

The Apple logo is a trademark of Apple Computer, Inc., registered in the U.S. and other countries. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

Every effort has been made to ensure that the information in this manual is accurate. Apple is not responsible for printing or clerical errors.

Apple Computer, Inc.  
1 Infinite Loop  
Cupertino, CA 95014-2084  
408-996-1010  
<http://www.apple.com>

Apple, the Apple logo, AppleScript, HyperCard, HyperTalk, Mac, Macintosh, PowerBook, Power Macintosh, QuickDraw, and QuickTime are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

Finder, MacinTalk, and Sound Manager are trademarks of Apple Computer, Inc.

SOJOURNER™, MARS ROVER™, and SPACECRAFT DESIGN AND IMAGES © 1996-97, California Institute of Technology. All rights reserved.

Other company and product names mentioned herein are trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

Simultaneously published in the United States and Canada.



# Contents

## **Preface**

### **Welcome 7**

Getting Started 7

System, Software, and Memory Requirements 8

Installing HyperCard 9

Learning About HyperCard 10

    If You're New to HyperCard 10

    If You're Already Familiar With HyperCard 11

    Updated Information 11

## **1 Assigning Button Tasks 13**

Opening the Button Tasks Window 13

Assigning Tasks 14

More About Tasks and Options 15

    Go to Destination 15

    Visual Effect 15

    Launch Application 15

    Link to URL 16

    Movie 16

    Sound 17

    Speak Text 18

    Other Features 18

## **2 Using QuickTime and QuickTime VR Features 19**

Opening Movie Files With the Movie Button Task 19

Opening Movie Files in the Finder 20

Keyboard Support of QuickTime VR Movies 20

QuickTime Tools Features 20

### **3 Using the New Syntax and Scripting Features 23**

Text to Speech 24

Sound 25

Clipboard 26

Number of Movies and References to Movies 27

Opening a URL 27

Opening Files 27

version Function 28

Picture XCMD 29

Apple Events 30

    picture 30

    Text Ranges 30

Script Editing Enhancements 31

### **4 Installing and Using Color Tools 33**

Installing Color Tools Into Your Home Stack 33

Turning Color Tools On and Off 34

Preparing a Stack for Color 35

Preparing a Color Stand-Alone Application 35

Getting Started With Color Tools 36

    Memory 37

    Using Color Commands in Scripts 37

    What's Next? 38

### **5 Color Editor Basics 39**

Opening the Color Editor 39

Understanding Overlays and Layers 40

Overview of the Color Editor Menus 40

    File Menu 40

    Edit Menu 41

    Items Menu 42

    Effects Menu 43

    Color Menu 44

Using the Color Palette	44
Color Palette Icons	44
Adding Color to Cards and Objects	45
Coloring a Card or Creating a Rectangle	45
Coloring Buttons and Fields	46
Using the Info Box	47
Adjusting Color	47
Adding Three-Dimensional Effects	48
Changing a Button or Field's Position	48
Changing a Button or Field's Size	48
Editing an Overlay	49
Using Menu Commands	49
Copying a Color From One Object to Another	49

## **6 Working With Pictures and Color Paint Tools 51**

PICT Resources Versus PICT Files	51
PICT Resources	51
PICT Files	52
Installing and Displaying a PICT Resource	52
Adding the PICT Resource to Your Stack	52
Placing the Image on a Card	53
Displaying a PICT File	54
Scaling PICT Images	55
Cropping PICT Images	56
Dithering	56
Preparing to Create a New PICT Image	57
Using the Color Paint Tools Palette	58
Choosing and Editing Patterns	59
About the Tools	59
Setting Effects	64
Paint Effects Shortcuts	68
Importing and Exporting Images	68
Importing an Image	68
Exporting an Image	68
Saving a PICT Image	69

Special Features of the PICT Editor 69

Editing Colors 69

Zooming 69

Creating a Gradient 70

Closing the PICT Editor 71

## **7 Scripting for Color 73**

Syntax Terms and Conventions Used in This Chapter 73

The addColor XCMD 74

addColor Structure 75

## **Appendix A Troubleshooting 127**

Color 127

Opening Files 128

Images in Stacks 129

Button Tasks 129

Visual Effects 130

Scripts for Play, Rewind, and Pause 130

Other 130

## **Appendix B Updates 131**

HyperCard 131

HyperTalk 132

## **Appendix C Quick Color Tutorial 135**

Making Sure Color Tools Are Installed 135

Preparing to Add Color 136

Opening the Color Editor 137

Adding Color to Buttons 137

Adding a Color Backdrop 138

Adding Color to Fields 139

Closing the Color Editor 140

Where to Go From Here 140

## **Index 141**

# Welcome

Congratulations on purchasing HyperCard! This manual provides instructions for features recently added to HyperCard, including

- prescribed button tasks you can use to launch applications or open movie files, connect to the Internet, assign speech and sound effects, or navigate to a new card—all at the click of a button
- a movie button task that opens more kinds of QuickTime media files—including JPEG, MPEG, and GIF
- the ability to use HyperCard to open movie and image files by dragging them onto the application icon in the Finder
- updated QuickTime movie and QuickTime VR commands, including commands to search for text in text tracks and apply interesting effects such as flipping, scaling, and rotating
- new syntax for controlling speech, sound, the Clipboard, and more
- new Picture external command (XCMD) capabilities
- new Apple Event capabilities
- an improved script editor window for easier navigation of scripts
- full-speed compatibility for all Power Macintosh computers
- Color Tools to add color to stacks

## Getting Started

To get started using HyperCard, read this preface first. The next two sections describe the system, memory, and software requirements and how to install HyperCard. The last section in this preface, “Learning About HyperCard” on page 10, describes the set of instructions you’ll use next.

**Important** Information in this preface supersedes any other information you may see printed in other manuals that came with HyperCard.

## System, Software, and Memory Requirements

HyperCard requires

- an Apple Macintosh, Power Macintosh, or PowerBook computer
- system software version 6.0.5 or later
- at least 2 megabytes (MB) of random-access memory (RAM); 4 MB of RAM required for System 7 or later; 8 MB of RAM required for Power Macintosh computers

Some features and components of HyperCard have additional requirements:

### QuickTime features

- To use QuickTime features, you must have a 68020 or later microprocessor and the QuickTime extension version 2.5 or later. Some new features require QuickTime 3.0 (included when you perform an Easy Install of HyperCard) or later.
- To use QuickTime VR features, you must have QuickTime VR 2.0 or later. QuickTime VR 2.1 is included when you perform an Easy Install of HyperCard.

### Button tasks and related HyperTalk commands

System 7 or later is required for button tasks.

Component Manager is required and included when you perform an Easy Install of HyperCard with System 7.0. Component Manager is a part of System 7.1 and later.

On Power Macintosh computers, button tasks require the ObjectSupportLib file. If you have System 7, this file is installed when you perform an Easy Install of HyperCard. The ObjectSupportLib file is a part of Mac OS 8.

Macintosh computers with 68000-series microprocessors require System 7.1 or later, or System 7.0 with the AppleScript or QuickTime extension (extensions included when you perform an Easy Install of HyperCard).

- The movie button task and the HyperTalk `movie` command require the QuickTime extension. (QuickTime 3.0 is recommended and included when you perform an Easy Install of HyperCard.)
- The text-to-speech button task and the HyperTalk `speak` command require Speech Manager software (included when you perform an Easy Install of HyperCard).
- The sound button task and the HyperTalk sound commands require system software version 6.0.7 or later and a computer with a 68020 or better microprocessor.
- The Link to URL button task and the HyperTalk `open URL` command require the Internet Config extension. Version 1.4 or later is recommended. Go to the World Wide Web site <http://quinn.echidna.id.au/Quinn/Config/> for more information about Internet Config, or download it from your favorite shareware server.

### **Color**

- To use Color Tools, you need a color-capable computer and 32-bit QuickDraw (included with System 7 and later; also included when you perform an Easy Install of HyperCard with system software version 6). The HyperCard memory partition must be set to at least 2200K, and 5 MB of RAM is recommended. On Power Macintosh computers, the memory partition should be set to at least 5120K and virtual memory should be turned off. The monitor's color depth should be set to display at least 256 colors or shades of gray.
- Color paint tools require System 7 or later.

### **Saving stacks as applications**

- System 7 is required to save a stack as an application.
- Component Manager is required and included when you perform an Easy Install of HyperCard on System 7.0. Component Manager is a part of the system software on System 7.1 and later.
- On Power Macintosh computers, the ObjectSupportLib file is required. If you have System 7, this file is installed when you perform an Easy Install of HyperCard. The ObjectSupportLib file is a part of Mac OS 8.

### **AppleScript**

To use AppleScript, you need the AppleScript software, which requires System 7 or later. Because of the differences between the AppleScript software for System 7 and Mac OS 8, the HyperCard Installer will only install AppleScript software if you're using System 7. If you have Mac OS 8 and are missing AppleScript software, you can install it from your system software CD-ROM disc.

## **Installing HyperCard**

To install your new software, follow these instructions:

- 1** Insert the *HyperCard* CD-ROM disc into the CD-ROM drive.
- 2** If the disc icon is not already open, double-click to open it.
- 3** Double-click the Read Me file to open it; then read its contents.

If the installation instructions in the Read Me or in the Installer differ from what you see here, follow the instructions in the Read Me or Installer.

- 4** Double-click the HyperCard Installer icon.
- 5** Read the licensing agreement. If you agree to its terms, click Accept.
- 6** If you want to install the HyperCard application and add new files, choose Easy Install. If you only want to install certain files, choose Custom Install and select the items you want to install or update.

- 7 Click Switch Disk to select a disk to install HyperCard on.
- 8 Click Install. When the Installer is finished installing the application, click Quit.
- 9 To install additional application programs packaged with this version of HyperCard, see the documentation that came with the programs.

## Learning About HyperCard

### If You're New to HyperCard

If you're using HyperCard for the first time, take the HyperCard Tour. You can access the HyperCard Tour by double-clicking the HyperCard application icon, then clicking the button labeled HyperCard Tour in the Home stack window that appears. The document "What is HyperCard?" also shows helpful information for beginners.

Next, you can read the *HyperCard Reference Manual* and this book, which describes HyperCard features added since the *HyperCard Reference Manual* was printed. If you want to learn how to create HyperTalk scripts (programs that allow more control and customization of your HyperCard documents), read the overview in Chapter 3 of the *HyperCard Reference Manual*, and then see the *HyperCard Script Language Guide* for more information. There is also a color tutorial (Appendix C, "Quick Color Tutorial,") scripting information on color, and some information on new syntax in this book.

As you become comfortable using HyperCard, you may find it more convenient to use the following online help documents, available by clicking buttons on your HyperCard Home card:

- HyperCard Help
- New Features
- QuickTime Tools documentation (part of the QuickTime Tools stack)
- HyperTalk Reference
- Audio Help
- HyperCard AppleScript Reference
- Apple Event Primer

You can also open HyperCard Help by pressing `⌘-?`, and you can open HyperTalk Reference by typing `HyperTalk` and pressing Return in the message box.

### **If You're Already Familiar With HyperCard**

Your main sources of information about the most recent features of HyperCard include this book, which describes HyperCard features added since the *HyperCard Reference Manual* was printed; the New Features stack; the documentation in the QuickTime Tools stack; and the HyperTalk Reference stack.

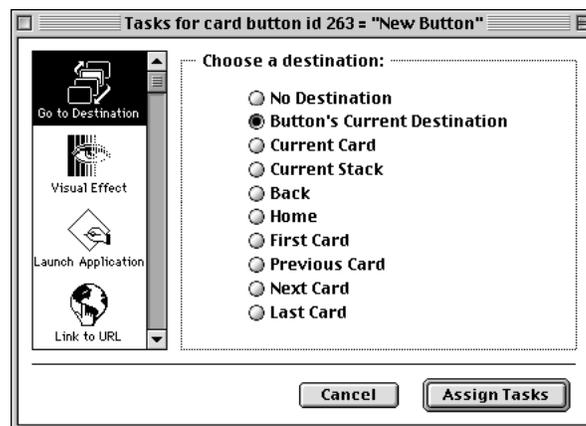
### **Updated Information**

This manual includes minor changes to instructions in your other HyperCard manuals, due to software changes. If you encounter instructions in your manuals that do not seem accurate, check Appendix B, "Updates," and the Read Me file (in the HyperCard folder after you've installed HyperCard) for updated information.



## Assigning Button Tasks

When you want to use a button in your stack to go to a new card, show a visual effect as it moves to a card, launch an application, link to the Internet, or play a movie, sound, or speech, you can assign built-in tasks to the button in the Button Tasks window. When you assign a task, a HyperTalk script is automatically generated for you.



**Note:** To use button tasks, you must be using System 7 or later. For additional requirements, see “System, Software, and Memory Requirements” on page 8.

### Opening the Button Tasks Window

To open the Button Tasks window, follow these steps:

- 1 Select the Button tool on the Tools palette.
- 2 Select the button you want to assign a task to, or create a new button by choosing New Button from the Objects menu.
- 3 Choose Button Info from the Objects menu (or double-click the button).  
A dialog box appears.

- 4 Click Tasks.

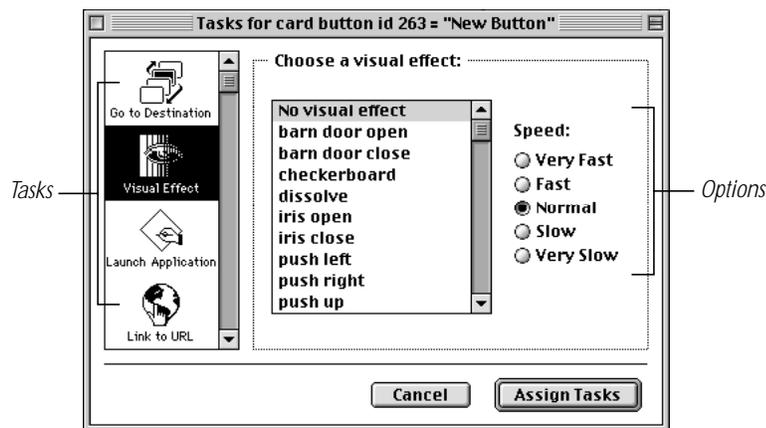
The Button Tasks window appears.

### Assigning Tasks

To assign a task to the button, follow these steps:

- 1 Click a task on the left side of the Button Tasks window.

The right side of the window displays options for that task.



- 2 Click the option or options you want to use.

Some features let you set more than one option, and some require that you enter more information. (For details, see the next section, “More About Tasks and Options” on page 15.)

- 3 Repeat steps 1 and 2 to assign additional tasks to the button.

- 4 When you've finished, click Assign Tasks.

HyperCard creates a script for the button that adds the tasks you want and closes both the Button Tasks window and the Info dialog box.

You can view the script that was generated for the button by choosing Button Info from the Objects menu again and clicking Script in the Info dialog box.

## More About Tasks and Options

Here's a guide on how to use each task and its options.

### Go to Destination

You use this task to navigate to a card in the Home stack or in a stack that is open on the screen. Each option for this task is described below.

Option	Action
No Destination	Removes any task already assigned that takes you to a card
Button's Current Destination	Maintains the current card destination
Current Card	Active card currently on the screen
Current Stack	First card of currently active stack
Back	Card you were on when you moved to the current card
Home	First card in the Home stack
First Card	First card in the current stack
Previous Card	Card before the current card in the current stack
Next Card	Card after the current card in the current stack
Last Card	Last card in the current stack

### Visual Effect

You can assign a visual effect to a button to make movement between cards and stacks more noticeable and visually interesting. You can choose both the kind of effect and its speed—how fast you want the effect to happen.

*If you're using color:* If you'll be leaving or going to a card with color information on it, don't use a visual effect. Instead, use a transition effect in the color editor. See Chapter 5, "Color Editor Basics," for more information.

### Launch Application

Use this task to open an application (and a document at the same time if you want).

To use this task, click the checkboxes next to "Launch application" and "With document" (if you want to open a document too), then click Choose to choose an application or document. (If nothing has been chosen yet, a dialog box automatically appears when you click the checkbox.) If you decide not to open the application or document, click the checkbox again to deselect it.

**Link to URL**

Use this task to launch the default World Wide Web browser (as defined in the Internet Config application) and connect to the URL (uniform resource locator) you specify. For this feature to work, the computer of the person using your stack must have the Internet Config extension installed and active in the System Folder, and the Internet Config application must have a Web browser specified in the Helpers window of Internet Preferences.

To use this task, click the Open URL checkbox to select it, and type a URL in the text box in the format `http://(address)`.

**Movie**

Use this task to open movie and image files. If you have the appropriate extensions, you can open GIF, JPEG, MPEG, and QuickTime VR files. For more information, see “System, Software, and Memory Requirements” on page 8, and Chapter 2, “Using QuickTime and QuickTime VR Features.”

To use this task, click the Play Movie checkbox to select it, then click Choose. (If nothing has been chosen yet, a dialog box automatically appears when you click the checkbox.)

You have several options for displaying the movie, described below.

<b>Option</b>	<b>Action</b>
Draggable Window	Displays a movie window with a title bar and a close box. You should choose this option if you want the user to be able to drag and close a QuickTime VR movie. For other movies, if you don't want the last frame to remain on the screen after it's finished playing, choose this option or Close After Playing.
Floating Window	Brings the movie window to the front of any other open windows.
Show Movie Controller	Displays a movie window with controls for viewing the movie.
Close After Playing	Automatically closes the movie window when it is finished playing. If you do not choose this option, you should choose Draggable Window if you want the user to be able to close the movie. (This option is not available for QuickTime VR movies.)
Centered on Screen	Displays the movie in the center of the screen.
Top Left Corner at	Displays the movie window at a location you enter in the text box, relative to the upper-left corner of the card window. Click the Set button and move your cursor around the screen to view the numerical values of different locations. Then click the screen at the desired location to set the value.

## **Sound**

See "System, Software, and Memory Requirements" on page 8 for this feature's requirements.

Use this task to play a sound when a button is clicked.

To use this task, you have several options:

- You can choose a prerecorded sound from the list on the left side of the Sound Button Tasks window.
- You can record a sound at three quality levels (the higher the level, the more disk space it uses) by clicking the Record Sound button under New Sound.
- You can import a sound that has already been recorded by clicking the Import Sound button.
- You can choose No Sound from the list in the left of the Sound Button Tasks window to play no sound when a button is clicked.

### **Speak Text**

See “System, Software, and Memory Requirements” on page 8 for this feature’s requirements.

Speech sounds best on faster computers, such as Power Macintosh G3 models. On older models with much lower processor speeds, some voices may be distorted.

Use this task to play text when a button is clicked.

To use this task you have several options:

- You can choose Button’s Current Speech Options to keep any currently assigned speech options.
- You can choose Speak Currently Selected Field to speak the text in a selected field. To use this option, follow these steps:
  - 1** Click on the card window.
  - 2** Choose the Field tool and select a field.
  - 3** Click the Tasks window. The button for Speak Currently Selected Field is now selected, and the ID of the field you selected is displayed beneath it.
- You can choose Speak String and type the text you want to have spoken into the text box.
- You can choose No Speech to play no sound when a button is clicked.

There are four options for the type of voice to speak the text: Default, Robotic, Female, or Male.

### **Other Features**

The tasks in the Button Tasks window are extendible. Software developers can supply you with new tasks, so you may see other tasks in the window.

## Using QuickTime and QuickTime VR Features

HyperCard now has several features for integrating QuickTime and QuickTime VR movies into stacks.

- You can have a movie open at the click of a button by using the movie button task.
- You can open and view movie files by dragging them onto the HyperCard application icon.
- The user of a stack can use the keyboard to navigate through QuickTime VR movies.
- You can control the display of movies and apply interesting effects with QuickTime Tools and the new QuickTime syntax.

### **Opening Movie Files With the Movie Button Task**

The Movie button task allows you to open any file that QuickTime 3.0 can open—including JPEG, MPEG, GIF, BMP, PNG, SGI, AVI, Photoshop, PICT, Targa, TIFF, FLC, and DV files—and display them in movie windows. If you have QuickTime VR 2.0 or later installed on your computer, you can also open and interact with QuickTime VR files.

**Note:** To open MPEG movies, you need the QuickTime MPEG extension version 1.0.1 or later. Certain other file types require QuickTime 3.0 or later. See “System, Software, and Memory Requirements” on page 8 for additional requirements.

See Chapter 1, “Assigning Button Tasks,” for information on how to assign movie tasks using the Button Tasks window.

## Opening Movie Files in the Finder

You can use HyperCard to open and play QuickTime movie files. You do this by dragging them onto the HyperCard application icon.



**Note:** You may need to rebuild your desktop after installing HyperCard to access this feature.

If Automatic File Translation is turned on in the Mac OS Easy Open control panel, you can also open other image files, such as PICT or GIF files, this way.

## Keyboard Support of QuickTime VR Movies

You or someone using your stack can control QuickTime VR movies by using the keyboard navigation keys. For example, pressing the Shift key will zoom in and pressing the Control key will zoom out.

You must click and hold the pointer in the VR movie window while pressing the Shift and Control keys to navigate.

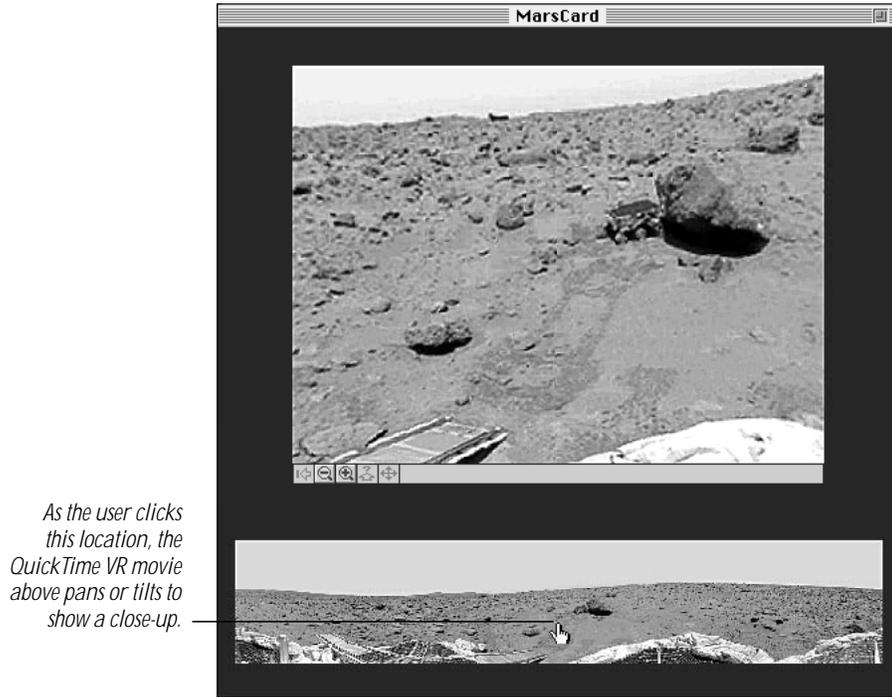
## QuickTime Tools Features

The QuickTime Tools stack allows you to use many new features of QuickTime and QuickTime VR. See the “Documentation” section of the QuickTime Tools stack for a complete explanation of the new features. The QuickTime Tools stack also has a toolkit to help you build scripts.

Using HyperTalk QuickTime syntax, you have precise control over the presentation of QuickTime movies in a HyperCard stack. You can adjust screen location, layer, window style, language, image quality, angle of view (for QuickTime VR movies), speed and direction, sound, and other characteristics. You can search for text in text tracks and apply such effects as flipping, scaling, rotating, and skewing. You can also specify the level of control someone using your stack has over the way a movie is viewed.

You can use HyperCard to integrate media in interesting ways. For example, you can play a movie while executing other commands in HyperTalk (such as playing an animation).

The illustration below shows HyperCard's unique capability to use one image or movie to control another. In this example, the lower JPEG image, a 360-degree panoramic still image, serves as a map to the QuickTime VR movie on the top. When the user clicks or drags across the still image, the movie pans or tilts to show a close-up view of the selected location.



*As the user clicks this location, the QuickTime VR movie above pans or tilts to show a close-up.*



## Using the New Syntax and Scripting Features

This chapter describes recently added scripting features or improvements, including

- HyperTalk syntax for controlling speech, sound, the Clipboard, number of movies, references to a specific movie that is open, opening a URL, and the version function
- ability to open more files at once
- new Picture external command (XCMD) capabilities
- new Apple Event capabilities
- enhancements to the script editing window that make it easier to navigate HyperTalk scripts

HyperTalk syntax has also been added to control QuickTime and QuickTime VR features and color. For an overview of QuickTime commands, see Chapter 2, “Using QuickTime and QuickTime VR Features.” See the “Documentation” section of the QuickTime Tools stack and the HyperTalk Reference for a complete explanation. For more information on color syntax, see Chapter 7, “Scripting for Color.” For information on changes to existing HyperTalk syntax, see Appendix B, “Updates.”

This chapter describes the following new HyperTalk words:

- `speak` translates a phrase into speech.
- `the speech` returns the text passed to the `speak` command.
- `stop speech` aborts the current and pending `speak` commands.
- `the voices` returns the voices available for generating speech.
- `the sound channel` determines the channel on which the next sound will be generated.
- `stop sound` aborts all sound on all channels.
- `the clipboard` is a container that holds the textual contents of the Clipboard.
- `the number of movies` returns the number of open movies.

- `first movie`, `last movie`, and `any movie` specify the open movie to which you are referring.
- `open URL` launches the default browser application and opens the World Wide Web URL you specify.

## Text to Speech

See “System, Software, and Memory Requirements” on page 8 for this feature’s requirements.

Speech sounds best on faster computers, such as Power Macintosh G3 models. On older models with much lower processor speed, some voices may be distorted.

### speak

```
speak phrase [with {genderLiteral voice | voice voice}]
```

```
speak "Hello, World."
```

```
speak field 3 with voice "Otis"
```

```
speak theSentence with female voice
```

*phrase* is any container or quoted string. *genderLiteral* is `male`, `female`, or `neuter`. *voice* is any voice in the Voices folder or in the MacinTalk file in the Extensions folder.

The `speak` command converts the text in *phrase* to speech, optionally using *voice*, and plays it through the device selected in the Sound control panel.

Speech is generated asynchronously and can be generated while HyperCard is in the background.

### the speech

```
the speech
```

```
speech()
```

```
put the speech into currentSpokenPhrase
```

```
if speech() is "done" then speak "That's all, folks"
```

The `speech` function returns the currently generated speech text. If there is no currently generated speech text, `speech` returns `done`.

### stop speech

```
stop speech
```

```
stop speech
```

```
if the time > "10:00 PM" then stop speech
```

The `stop speech` command stops the current speech and aborts pending speech commands. You can also stop the current and all pending speech, as well as all current and pending sound, by pressing ⌘-period.

#### the voices

```
the voices  
voices()
```

---

```
put the number of lines in the voices into voiceCount  
speak "Who's there?" with voice (any line of the voices)
```

---

The function `the voices` returns a return-delimited list of the voices currently available for generating speech.

For this function to work properly, all voices must be either in the Voices folder or in the MacinTalk file in the Extensions folder.

## Sound

See “System, Software, and Memory Requirements” on page 8 for this feature’s requirements.

HyperCard supports up to eight channels for playing sounds. Channels can be played concurrently or one at a time.

**Important** To use this feature, you need system software version 6.0.7 or later operating on a computer with a 68020 or higher microprocessor. Some computers with 68020 or 68030 microprocessors may use fewer than eight sound channels.

#### the soundChannel

```
set [the] soundChannel to integerValue
```

---

```
set soundChannel to value(the soundChannel) + 1  
if the soundChannel = 1 then play theTune
```

---

*integerValue* resolves to a whole number in the range 1 through 8.

The `soundChannel` property is the channel through which sound is played. The sound must have been generated by the `play` command.

The `play` command operates on the current sound channel. By immediately switching channels and playing new sounds, several sounds can be played nearly simultaneously.

```
on chord
  play harpsichord C E G
  wait 2 seconds
  set soundChannel to 1
  play harpsichord C
  set soundChannel to 2
  play harpsichord E
  set soundChannel to 3
  play harpsichord G
end chord
```

#### **stop sound**

```
stop sound
```

---

```
stop sound
```

```
if the voice is "Manager" then stop sound
```

The `stop sound` command stops the current sound and aborts pending sound commands on all channels.

To stop the current and all pending sounds plus all current and pending speech text, press ⌘-period.

To stop the sound on the current sound channel only, use `play stop`.

## **Clipboard**

`clipboard`, a new container, reflects the text contents of the Clipboard.

#### **the clipboard**

```
get [the] clipboard
```

```
put textValue into the clipboard
```

---

```
if the clipboard contains "Fred" then doMenu "Paste Text"
```

```
put the date && the time into the clipboard
```

**Important** The expression `the clipboard` is guaranteed to give satisfactory results only when HyperCard is the active application. To transfer information between applications, use AppleScript variables.

## Number of Movies and References to Movies

The expression `the number of movies` returns the number of open movie windows.

The expression `first movie` refers to the same movie as movie 1. If there are three open movie windows, `last movie` refers to movie 3. These ordinals function just as they do when referring to buttons, fields, cards, and so on. See “Chunk Expressions” in the *HyperCard Script Language Guide* for more information.

### the number of movies

```
the number of movies
```

```
{first | last | any} movie
```

---

```
if the number of movies >1 then close movie 2
```

```
if the name of first movie is "MyHomeMovie" then open movie  
"BackdropMovie"
```

---

## Opening a URL

Use the `open URL` command to launch the default Web browser (as defined in the Internet Config application) and go to the URL (uniform resource locator) you specify. For this feature to work, the computer of the person using your stack must have the Internet Config extension installed and active in the System Folder, and the Internet Config application must have a Web browser specified in the Helpers window of Internet Preferences.

### open URL

```
open URL
```

---

```
open URL http://www.apple.com/hypercard
```

---

## Opening Files

You are no longer limited to a maximum of three open files at one time when using the `open file` command.

When you use this command with QuickTime 3.0 or later and you create a new file with any standard suffix used by QuickTime (such as “.gif” or “.jpeg”), the file will be changed to take the corresponding type and creator, and its icon will reflect this. For instance, files with the suffix “.gif” will show a GIF icon.

To prevent file icons from being changed, open the QuickTime Settings or QuickTime control panel and deselect “Enable QuickTime Exchange,” then restart your computer.

## version Function

Previously a property, `version` is now a function that returns version information for the HyperCard application, a HyperCard stack, the Mac OS, AppleScript, Sound Manager, QuickTime, and QuickTime VR.

Except when used to find the version of a stack, the version of Mac OS, or the version of AppleScript, the `long version` returns an 8-digit number specifying the major version number, the minor version number, the software state (for example, beta), and the release number.

When used to find a stack's version, the `long version` returns a list of five comma-separated 8-digit numbers. the `long version of Mac OS` returns an 8-digit number, zero-filled to represent the version as an integer (example: 00000810 for Mac OS 8.1).

When used to find the version of AppleScript, the `long version` returns an 8-character hexadecimal string that represents the version of its application programming interface, or API (first four digits), and the version of the software (last four digits). For AppleScript 1.1, the hex string is 01100110. If you do not use the option `long`, this function returns the version as a decimal string. For AppleScript 1.1, the decimal string is 1.10.

You can compare the returned value to an arbitrary number. For example:

```
if the version of "AppleScript" >= 1.1 then...
```

See pages 494–495 in the *HyperCard Script Language Guide* for more information.

**Note:** It is no longer necessary to use `scriptingLanguage` when referring to AppleScript.

### version

`version`

`long version`

---

the [long] version of Mac OS

the [long] version of HyperCard

the [long] version of this stack

the [long] version of AppleScript

the [long] version of Sound Manager

the [long] version of QuickTime

the [long] version of QuickTime VR

---

## Picture XCMD

HyperCard now uses QuickTime graphics importers for opening file types other than PICT or PNTG if QuickTime 2.5 or later is installed. Using the Picture XCMD, you can open all image file types that QuickTime recognizes. Unlike using Movie to open a QuickTime-compatible image, Picture XCMD allows you to open an image without displaying it in a movie window.

The Picture XCMD can now accept as a parameter PICT data, a file name, the Clipboard (when it contains PICT data), or the name of a PICT resource. Pass the string `appleEvent` as the second parameter and pass the picture data of type PICT in any later parameter, as the following AppleScript example shows.

```
on mouseUp
    tell application "Microsoft Excel"
        make Document
        copy "=sin(row()/8)*cos(column()/4)" to ~
        first Cell of first Row of first Document
        copy first Cell to Range "R1C1:R20C10"
        make Chart
        copy {10,60,600,400} to bounds of first Window
        copy three D Surface to type of first Chart
        copy first Chart to theChart
    end tell
    |picture|("Excel chart", "appleEvent", theChart)
end mouseUp
```

You can use the Picture XCMD within an `appleEvent` handler to display the PICT image contained in the direct parameter of an Apple event. Use the following form:

```
on appleEvent class,id,sender
    if class & id is "some custom event class and id"
        picture "name", "appleEvent", ...
    else pass appleEvent
end appleEvent
```

This works for events that contain pictures in their direct parameters, but not for events that contain aliases to PICT files, such as `oDoc` events.

## Apple Events

### picture

The property `picture` is a new OSA (Open Scripting Architecture), read-only property of card and background objects that returns the bitmap of a card or background as a QuickDraw picture. It lets you pass around picture data so that you no longer need to use the paint tools to select and copy a picture.

This AppleScript example puts the background picture into a PhotoBook document:

```
tell application "HyperCard"
    copy picture of current background to BkgndPaint
end tell

tell application "PhotoBook"
    activate
    set clipboard to BkgndPaint
    paste
end tell
```

### Text Ranges

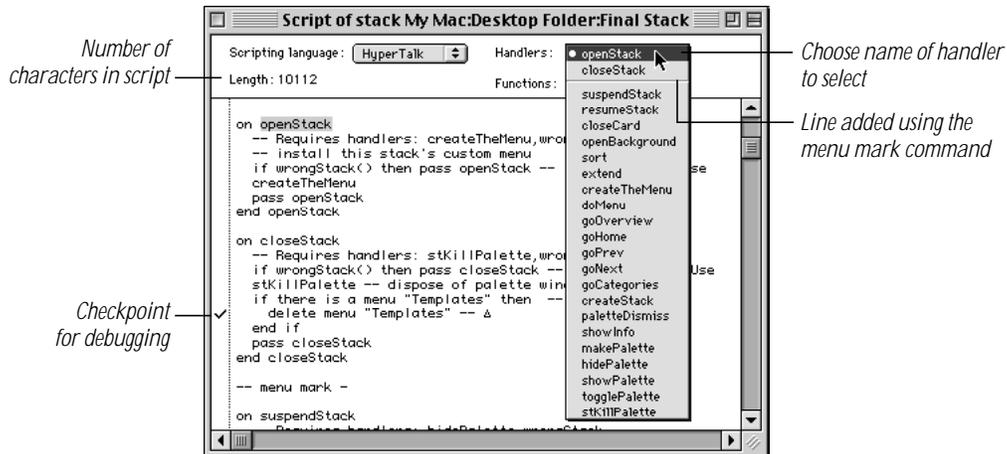
HyperCard supports negative indices for text ranges in OSA scripting languages.

Negative indices are interpreted as offsets from the end of the text. So `character -1 of card field 1` in AppleScript is the last character of card field 1.

## Script Editing Enhancements

The new script editor window makes it easier to search and navigate through HyperTalk scripts.

The top of the window shows the length of the script (number of characters). You can choose a type of handler or function and scroll to its location in the script. You can also click the side column to place checkpoints where you want a script to stop for debugging.



For added ease in navigation, follow these tips:

- To see names of handlers and functions sorted alphabetically, Shift-click the Handlers and Functions pop-up menus.
- To see a list of handlers and functions with the names of functions italicized, press the Command (⌘) key and click anywhere in the script editor window. (To see the list sorted alphabetically, press ⌘-Shift and click.)
- To have a line appear before or after the name of a handler or function in the Handlers or Functions pop-up menus, type `-- menu mark -` before or after the handler or function in the script.
- To have names of items in the Handlers and Functions pop-up menus appear differently, type the following text before the handler or function in the script:
  - `-- menu mark` followed by
    - <I to view the menu item in italic
    - <B to view the menu item in bold

See *Inside Macintosh*, a programming guide available at most software book retailers, for the entire list of options for changing the appearance of menus.



## Installing and Using Color Tools

The Color Tools allow you to add color to your stacks. This chapter describes how to prepare your Home stack so that you can use the Color Tools, how to turn Color Tools on and off, and how to add Color Tools to a stack. This chapter also provides an overview of using color to help you get started.

**Important** To work effectively with Color Tools, HyperCard's preferred memory size should be set to 2200K or higher (the higher, the better). On Power Macintosh computers, the preferred memory size should be set to 5120K or higher. See "System, Software, and Memory Requirements" on page 8 for additional requirements.

### Installing Color Tools Into Your Home Stack

To use Color Tools, you have to install them into your Home stack.

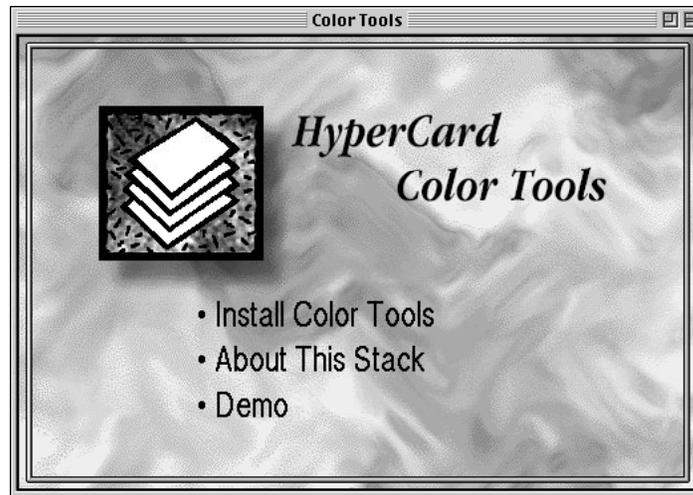
- 1 Be sure that the Color Tools stack is in the same folder as the HyperCard application.
- 2 Open the Color Tools stack.

Click the Color Tools button on the Stack Kit card of the Home stack, or double-click the Color Tools stack icon in the HyperCard folder.



The Color Tools screen appears.

- 3 Click Install Color Tools.



- 4 When the Install card appears, click Install.

(If you don't see the Install button, but instead see a button labeled "Remove," Color Tools are already installed.)

When the installation is complete, the Install button changes its name to "Remove." If you want to remove the Color Tools from the Home stack, you can return to this stack and click Remove.

*What Install does:* Clicking Install makes changes to the Home stack script, adds the Color menu to the menu bar, and installs a new button on the first card of the Home stack.



**Color Tools are ON**

You can move the "Color Tools are ON" button to a different place on the Home card if you like.

### **Turning Color Tools On and Off**

When you first install the Color Tools, HyperCard adds the Color Tools stack to its "stacks in use" list. This allows other stacks to take advantage of the scripts and resources in the Color Tools stack.

To shut off Color Tools, click the “Color Tools are ON” button. The Color menu disappears from the menu bar, the button changes to “Color Tools are OFF,” and stacks no longer have access to the Color Tools scripts and resources.



**Color Tools are OFF**

*Shutting off Color Tools doesn't shut off color:* Turning off Color Tools won't remove color already displayed in a stack. Such color is permanently part of the stack.

### **Preparing a Stack for Color**

To give a stack color capabilities, follow these steps:

- 1** Go to the stack to which you want to add color.
- 2** Make a copy of the stack using the “Save a Copy” command from the File menu.  
Conflicts occasionally occur between the resources in a stack and the resources that Color Tools add. Keeping an original copy of the stack makes it easy to recover from any problems.
- 3** Choose Open Coloring Tools from the Color menu.  
A dialog box appears, asking you to confirm that you want to add scripts and resources to your stack.
- 4** Click OK.

When the installation is complete, the color editor opens.

The color editor is described in Chapter 5, “Color Editor Basics.” The color paint tools are covered in Chapter 6, “Working With Pictures and Color Paint Tools.”

Installing Color Tools increases the size of your stack by 34K and introduces free space into the stack. You can reduce the stack's size by choosing Compact Stack from the File menu.

### **Preparing a Color Stand-Alone Application**

If you want to save a stack as a stand-alone application and you have added color resources to the stack, you need to close and reopen it before you save it.

Once the stack is saved as a stand-alone application, its memory allocation is the same as your version of HyperCard. You may have allocated extra memory to HyperCard that your application doesn't need. To reduce the memory allocated to the application, follow these instructions:

- 1** Make sure that the stand-alone application is not open.

You can't change the memory allocated to an open application.

- 2 Locate the stand-alone application.



### My Color Application

- 3 Click the application's icon once to select it.
- 4 Choose Get Info from the File menu.

You can also press **⌘-I**.

A dialog box appears.

- 5 Set the minimum memory size and the preferred memory size to 2200K (5120K for applications that may be used on Power Macintosh computers).
- 6 Click the close box.

## Getting Started With Color Tools

You can add color to any object in your HyperCard stack: buttons, fields, rectangles, and PICT images. You can also create your own color PICT images using color paint tools.

Keep in mind the following information about disk space and speed (the time it takes to redraw a card when moving through a stack) when choosing the objects you would like to color:

*Rectangles:* These are the most efficient objects to color. Color rectangles use up very little disk space, and redraw speed is fast.

*Buttons and Fields:* Adding color to these objects takes up very little disk space. Redraw speed is medium.

*Pictures:* Color PICTs take up considerable disk space, though draw-style PICTs are smaller than bitmapped PICTs. The redraw rate for larger PICT images can be slow. Use the crop tool to eliminate unnecessary border space in a PICT image. For the best performance, use compressed QuickTime pictures instead of PICT images.

*Color depth and speed:* The color depth of your monitor also affects redraw speed. For faster performance, choose lower color depths.

*Number of objects and colors:* It's best to limit the number of colors and colored objects you use in a stack. If objects don't change between cards, use a single card-sized picture; this will prevent the need to redraw several objects.

**Important** Adding color to your Home stack is not recommended. If not scripted correctly, a Home stack with color will cause HyperCard to run more slowly.

### **Memory**

Color also has a significant effect on the use of memory. To conserve memory when displaying color, follow these suggestions:

- Display color at a lower color depth.
- Reduce the size of your cards.
- Use smaller pictures.
- Open only one color stack at a time.

### **Using Color Commands in Scripts**

If your stack is controlled by a HyperTalk script, read the information below for additional tips on using color efficiently in stacks.

See Chapter 7, "Scripting for Color," for general information on new HyperTalk commands.

#### **Changing a card**

Each time you change a card with color by showing or hiding it or by moving color elements, the card must be recolored with a call to `addColor colorCard` or some other appropriate command. Recoloring the card is time-consuming.

If you must add color to the card, consider using temporary colors (that is, use `colorField` instead of `addField`).

Changes such as adding text to a field also slow the stack because they require calls to the Color Tools external command (XCMD). The fewer such calls you make, the faster your stack operates.

Whenever possible, use `colorCard` instead of `colorCardLayered`. (`colorCardLayered` takes twice as long to execute.)

If your color information doesn't change, embed the values in your scripts.

#### **Moving from card to card**

Whenever possible, remove the `addColor` calls in `openCard` handlers. For example, if you have no card layer color and no changing color buttons or fields, you need to draw color only when you open the background. (However, you won't get a transition effect when you change cards.)

### Coloring existing stacks

When you open the color editor for the first time in any stack, a number of calls are added to that stack's script. To color existing stacks by hand, use the following calls in their respective system message handlers:

Call	Handler
install	openStack
colorCard	openCard
lock screen	closeCard
remove	closeStack
pass <i>messageName</i>	<i>all handlers</i>

### What's Next?

Go to the next chapter, Chapter 5, "Color Editor Basics," to learn how to use the color editor, and how to color rectangles, fields, and buttons.

## Color Editor Basics

You use the color editor to add or change colors in a stack and to manipulate color PICT resources. This chapter presents a brief overview of the editor. Topics include

- color editor menus and commands
- color transition effects
- the color palette
- adding color to cards and objects

See Chapter 6 for information on using PICT resources and files and creating PICT images. See Chapter 7 for information on scripting for color. See Appendix C for a quick tutorial on using color.

*Create a Practice stack:* To learn the most about using Color Tools, create a Practice stack with several fields and buttons in both the card and background to experiment with as you go through this and the next several chapters. See Chapter 4, “Installing and Using Color Tools,” for instructions on adding Color Tools to your Practice stack.

### Opening the Color Editor

To open the color editor, choose Open Coloring Tools from the Color menu.

If this is the first time you’re opening the editor in this stack, a dialog box appears, asking you to confirm that you want to add Color Tools to the stack. Click OK.

Any open palettes disappear, and the color editor opens. Color editor menus and the color palette appear.



*Closing the palette closes the editor:* Clicking the close box on the color palette closes the color editor and returns you to the standard HyperCard environment.

## Understanding Overlays and Layers

When you're in the color editor, you create or manipulate color overlays. A *color overlay* is any of the following: a PICT image displayed in the stack by using a color editor command, a color rectangle, or an element that adds the appearance of color to a button or field.

All color exists in a layer behind all paint pictures, buttons, and fields. The color layer holds all the color elements—button overlays, field overlays, rectangles, and PICT images. Color looks as if it's on top of other elements because the Color Tools make all white areas transparent (including opaque fields), allowing the color to show through. You can think of color as a piece of tinted cellophane laid behind a transparent object and cut to fit it.

## Overview of the Color Editor Menus

When you open the color editor, you enter a new environment with a new set of menus and commands.

The following tables list the commands and describe their results.

### File Menu

Command	Action
Quit HyperCard	Closes the color editor and quits HyperCard

## **Edit Menu**

<b>Command</b>	<b>Action</b>
Undo	Undoes a color change, a size change to a PICT image, or a move of any kind
Cut	Removes a color overlay (but not the object with which the overlay is associated)
Compact Color Database	Eliminates color information about deleted objects
Sort Color Database	Associates color overlays with their objects; arranges color buttons and fields in front of color rectangles and PICT images
Background	Opens the background for editing or adding color overlays

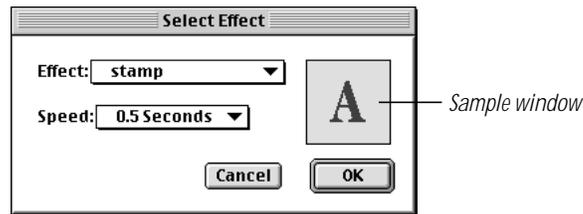
## Items Menu

Command	Action
Bring To Front	Moves the selected color overlay (but not its object) in front of other color overlays; analogous to ⌘-Shift-+ (plus) for a selected button or field
Bring Closer	Moves the selected color overlay (but not its object) one layer closer to the front; analogous to the Object menu's Bring Closer command for a selected button or field
Send Farther	Moves the selected color overlay (but not its object) one layer back; analogous to the Object menu's Send Farther command for a selected button or field
Send To Back	Moves the selected color overlay (but not its object) behind other color overlays; analogous to ⌘-Shift- - (minus) for a selected button or field
Place Picture	Displays a PICT image stored as a resource in this or any stack; retrieves and displays a PICT image from a file
Create New Picture	Enters the PICT editor so you can create a new color PICT image
Edit Picture	If a PICT image is selected, opens the PICT editor and displays the selected PICT image in an editing window; otherwise, opens the PICT editor and displays an empty editing window
Place Rectangle	Creates a new rectangle in the center of the screen using the currently selected color
Item Info (Get Info)	Shows information about the selected color overlay; presents editing options

## Effects Menu

Command	Action
Stack Transitions	Sets the default transition effect and duration for any card that doesn't already have a transition effect
Background Transitions	Sets the default transition effect and duration for the current background; background transitions take precedence over stack transitions
Card Transitions	Sets the transition effect and duration for the current card; takes precedence over all other transition settings

As you change settings, the sample window shows what the effect will look like.



**Note:** The transition effect dialog box does not indicate the name of the current effect.

The speed setting is relative to the computer you are using. The effect may take longer than shown in the sample window. Card changes take longer in color stacks than they do in black-and-white stacks.

### Why standard visual effects don't work

Commands in the Effects menu determine the transition effect in color stacks. The color editor adds a `closeCard` HyperTalk handler with a `lock screen` command to the stack. This command prevents HyperCard's standard visual effects from working as you change cards.

You can remove the `lock screen` command, but since HyperCard's visual effects work only on black-and-white images, if you follow a visual effect with any color effect, the image stutters as color is turned on.

## Color Menu

Command	Action
Open Coloring Tools	Opens the color editor; on first use, modifies the current stack script and adds color resources to the stack
Redraw Screen	Updates the display and realigns color overlays with the objects to which they belong (especially useful when you resize or move color buttons or fields)

## Using the Color Palette

The color palette appears when you open the color editor. You use the color palette to choose a color for the currently selected shape, button, or field. It also provides shortcuts for certain menu commands.

### Color Palette Icons

The icons at the top of the color palette enable different sets of tools and execute menu commands.



The following table lists the effects of clicking each icon.

Icon	Effect
Button	Enables selection of button overlays
Field	Enables selection of field overlays
Pict	Enables selection of PICT images; double-clicking initiates the process of displaying a PICT image (a shortcut for the Items menu's Place Picture command)
Rect	Enables selection of color rectangles; double-clicking creates a new color rectangle in the current color (a shortcut for the Items menu's Place Rectangle command)
Paint	Opens the PICT editor to edit the selected PICT image (a shortcut for the Item menu's Edit Picture command) or to create a new one (a shortcut for the Items menu's Create New Picture command)

## Adding Color to Cards and Objects

You add color to an entire card by creating a color rectangle overlay and sending it to the back. You color buttons and fields by adding overlays to them in the color editor. You can also add three-dimensional effects to any overlay.

When you edit a color overlay, the object itself remains unaffected. You can resize and move an object, but in order to make any other changes, you must quit the color editor.

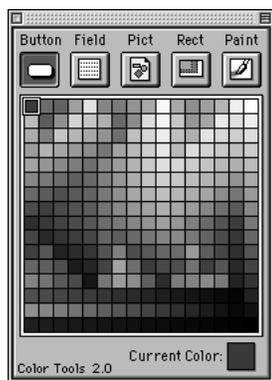
An object must already exist in the proper layer (card or background) before you can color it in the editor.

### Coloring a Card or Creating a Rectangle

To add a color overlay to a card, or to create a rectangle shape, follow these steps:

- 1 Choose Open Coloring Tools from the Color menu.

The color editor opens.



- 2 Click a color on the color palette.

If the palette doesn't include the color you want, double-click the current color at the bottom of the palette to open the color picker, then select a color. The color picking systems available to you depend on the color picking extensions installed on your computer.

- 3 Choose Place Rectangle from the Items menu, or double-click the Rect icon at the top of the color editor window.

A rectangle appears in the center of the card in the color you've selected.

- 4 Drag the edges of the rectangle to adjust the shape, and drag the rectangle until it is in the position you want.
- 5 If you want the rectangle to appear as a background color and you have other color objects on the card, send the rectangle to the back. To do this, make sure the rectangle is still selected and choose Send To Back from the Items menu.

## Coloring Buttons and Fields

To add a color overlay to a button or field, follow these instructions:

- 1 If necessary, create the object on the card or background layer.
- 2 Choose Open Coloring Tools from the Color menu.

The color editor opens.

- 3 Click the button or field icon at the top of the color palette.



- 4 Click the object you want to color.

You can color only one object at a time.

Clicking an object selects it. If you click the Button icon, you can select only a button; if you click the Field icon, you can select only a field.

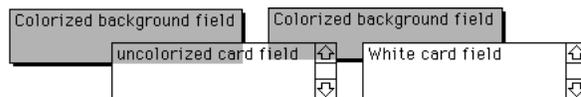
*How to find invisible objects:* To find borderless uncolored buttons in the color editor, press Shift-Option. To find borderless uncolored fields, press ⌘-Shift-Option.

- 5 Click a color on the color palette.

If the palette doesn't include the color you want, double-click the current color at the bottom of the palette to open the color picker, then select a color. The color picking systems available to you depend on the color picking extensions installed on your computer.

The selected object changes to the color you've selected. To avoid obscuring text, use light colors for fields and for buttons whose names show.

All HyperCard buttons and fields become transparent as soon as you add color resources to the stack. If you want an opaque object, you can color it white.



## Using the Info Box

You can use an overlay's Info box to

- adjust the color and open the color picker
- set or change a three-dimensional effect
- change the position of the object and overlay
- change the size of the object and overlay

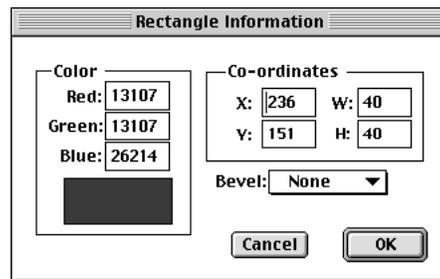
You can also control all of these settings in HyperTalk. See Chapter 7, “Scripting for Color,” for details.

To open the overlay's Info box, follow these instructions:

- 1** If the color editor isn't already open, choose Open Coloring Tools from the Color menu.
- 2** Click the Button, Field, or Rect icon, as appropriate, at the top of the color palette.
- 3** Click the color object you want to work with and choose Item Info from the Items menu (or double-click the object).

You must color an object before you can use its Info window.

The Info box opens, with the ID of the object in the title.



## Adjusting Color

- To open the color picker from the Info box, click the color field at the bottom of the Info box's Color window.
- To modify an RGB number, select it and type the new number.

You can immediately see the change in the color sample. However, some changes in RGB numbers are too subtle to see, especially if your monitor is set to display 256 colors or fewer. For more apparent changes, use the two leftmost digits.

### Adding Three-Dimensional Effects

You use the Bevel pop-up menu to add shading to a selected object.

You can add bevels of 1 to 6 pixels—the higher the number, the wider the bevel and the heavier the shadowing.



Transparent objects don't show bevels. To shade a transparent object, color it white first.

#### Bevel tips

- Heavy bevels can obscure text, especially in fields. To avoid this problem, turn on the Wide Margins option in the field's Info dialog box or choose a lower bevel number.
- Avoid bevels on color checkboxes and radio buttons.
- Oval buttons with a 3-pixel bevel look great with pastel colors.

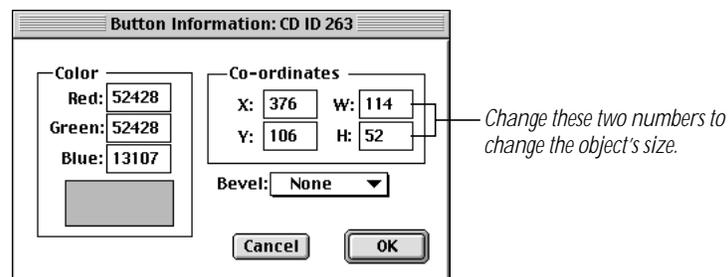
### Changing a Button or Field's Position

The easiest way to change an object's position is to drag it. You can position it more precisely, however, by setting its coordinates in the Info box's Co-ordinates window.

The X and Y coordinates represent a point X pixels to the right of and Y pixels down from the upper-left corner of the stack window. When you close the Info box, the upper-left corner of the object moves to the specified coordinates.

### Changing a Button or Field's Size

The easiest way to change a selected object's size is to drag its corner. You can precisely set its width and height in the Info box.



- To keep the object's proportions constant, change both numbers by the same percentage.

## **Editing an Overlay**

You edit an overlay the same way you create one. Follow these steps:

- 1** If the color editor isn't already open, choose Open Coloring Tools from the Color menu.
- 2** Click the Button or Field icon at the top of the color palette.
- 3** Select the object whose overlay you want to edit.
- 4** Make changes as appropriate.

Choose any menu item or select a new color on the color palette.

## **Using Menu Commands**

In the color editor, the Undo command in the Edit menu undoes the most recent move, resizing, or color change for the selected color object. There is no Redo command.

The Cut command (or the Delete key) removes the color from the selected object. It does not remove the object itself.

## **Copying a Color From One Object to Another**

The Copy command places the color and bevel of the selected overlay on the Clipboard. You can use the Paste command to assign that color and bevel to any other object of the same type. You cannot copy the color and bevel of one kind of object and paste it to another kind.

You can also write scripts to change colors. See Chapter 7, "Scripting for Color."

*Copying an object does not copy its color:* When you copy and paste (or drag-copy) a color button or field in HyperCard, the object's color is not copied. When you copy and paste or duplicate and paste a card, the colors associated with that card aren't pasted onto the new card.



## Working With Pictures and Color Paint Tools

This chapter describes how to display high-resolution PICT files and resources, properly scaled and cropped, anywhere on a card or background. It also covers how to create and edit PICT images.

*You can use scripts to display PICT images:* HyperCard includes several PICT scripting commands. See the sections on the `addPict`, `colorPict`, `addPictFile`, and `colorPictFile` commands in Chapter 7, “Scripting for Color,” for details. These commands complement the existing `Picture` command, described in the *HyperCard Script Language Guide*.

**Note:** The Color Tools can display only the system palette colors. If your PICT image was created using different colors, they will be converted to the system colors.

### **PICT Resources Versus PICT Files**

A PICT resource is the information stored in the resource fork of a file that describes picture information. A PICT file is the standard Macintosh image format.

In HyperCard, PICT resources reside in the stack in which they are displayed. PICT files are separate from the stack that displays them.

### **PICT Resources**

Because PICT resources are a part of the stack, they move with the stack when you copy or move it.

PICT resources increase a stack’s size. A stack with several PICT resources often doesn’t fit on a high-density floppy disk.

## PICT Files

If you display PICTs in your stack from PICT files, the stack and its PICT images are easier to transport, because you can use a compression utility to make the PICT files smaller.

You need to keep all the PICT files with the stack when you move it. The PICT files must go in the same folder as HyperCard, Home, or your stack.

## Installing and Displaying a PICT Resource

To use a PICT resource in your stack, you must first add it to your stack, then place the image where you want it on the card.

### Adding the PICT Resource to Your Stack

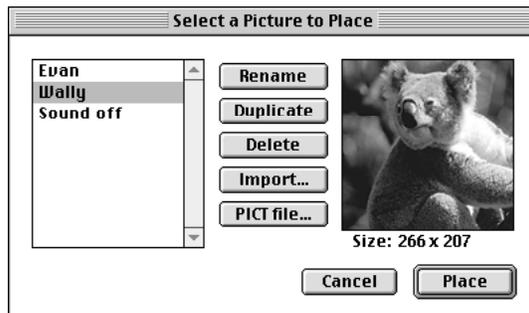
To add a PICT resource to your stack, follow these steps:

- 1 Choose Open Coloring Tools from the Color menu.
- 2 Choose Place Picture from the Items menu.

Or double-click the Pict icon at the top of the color palette.

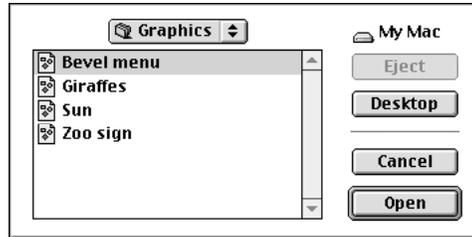


A dialog box appears with a list of the PICT resources already installed in the stack.



- 3 Click Import.

A dialog box appears.



- 4 Locate the PICT file whose resource you want to add and click Open. Or if the PICT resource is in another stack, locate the stack and click Open.

**Note:** PICT resources whose names start with a bullet (•) don't appear in the list and cannot be imported.

- If you opened a PICT file, a copy of the PICT image from the file is stored in your stack. (The original remains on the disk.)
  - If you opened a stack, a dialog box appears with a list of PICT resources you can copy to your stack.
- 5 If you opened a stack, select the PICT resource you want and click Import to copy the resource to your stack. When you're finished, click Done.

You return to the "Select a Picture to Place" dialog box. The name of the PICT resource you've imported appears in the list.

- 6 When you've finished adding PICT resources, click Cancel to close the dialog box and return to the color editor.

### Placing the Image on a Card

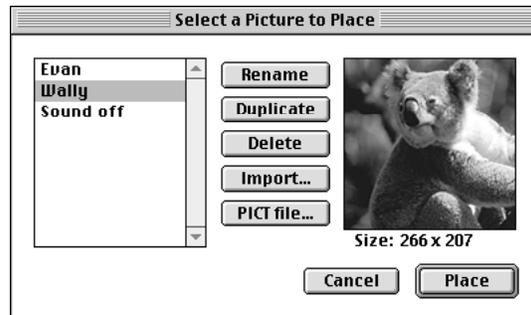
You place a PICT resource image on a card by using the "Select a Picture to Place" dialog box.

- 1 Open the stack in which you want to display an image.
- 2 Be sure that you're in the layer (card or background) where you want the PICT image to appear.
- 3 With the color editor open, choose Place Picture from the Items menu.

Or double-click the Pict icon at the top of the color palette. The "Select a Picture to Place" dialog box appears with the names of all the PICT resources contained in the current stack.

- 4 Click the name of the PICT image you want to show.

A miniature version of the PICT image appears in the dialog box's preview window, with its size in pixels. (The image may appear distorted.)

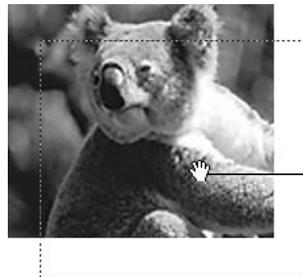


- 5 Click Place.

The PICT image appears on the screen.

- 6 Drag the PICT image where you want it.

Be sure to drag it by its middle. (Dragging a PICT image by its edges changes its proportions.) The pointer changes to a hand to indicate that you'll be moving the PICT image.



*When you see the hand,  
it's safe to drag.*

### Displaying a PICT File

When you use a PICT file in a stack, the file remains separate from the stack, and the stack displays the file by pointing to the file. This makes the stack smaller and allows you to compress PICT files when transporting stacks and associated images.

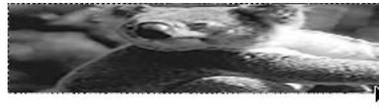
PICT files or aliases for them must be in the same folder as the HyperCard program, the Home stack, or your stack. If you use an alias for a PICT file, its name must be the same as the name of the original file.

To use a PICT file in your stack, follow these steps:

- 1** Be sure that you're in the layer (card or background) where you want the PICT image to appear.
- 2** With the color editor open, choose Place Picture from the Items menu.  
Or double-click the Pict icon at the top of the color palette. A dialog box appears.
- 3** Click PICT File.  
Another dialog box appears.
- 4** Find the PICT file you want to display.
- 5** Click Open.  
The PICT image appears on the screen.
- 6** Drag the PICT image where you want it.  
Be sure to drag it by its middle.

### **Scaling PICT Images**

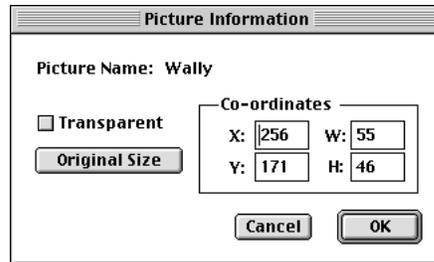
Scaling a PICT image (either a resource or a file) changes its height, width, or both. You can change a selected image's proportions by dragging one of its edges in the stack window.



You can scale a PICT image more precisely within its Info box. You can also scale PICT images from within the PICT editor. See "Scale" on page 66.

- 1** With the color editor open, click the Pict icon at the top of the color palette.
- 2** Click the PICT image you want to scale and choose Get Info from the Items menu.  
Or double-click the PICT image.

The PICT image's Info box appears.



- 3 Type new numbers into the Width (W) and Height (H) boxes.

To keep the proportions constant, change the W and H numbers by the same percentage.

- 4 Click OK.

The image changes after you close the Info box.

- To restore the image to its original size and proportions, click Original Size in the PICT image's Info box. You can restore the original size even if you've changed the PICT image's dimensions by dragging its corners.

## Cropping PICT Images

To crop a PICT image (either a resource or a file), you use the marquee tool from the paint tools palette.

- 1 Click a PICT image to select it.
- 2 With the color editor open, click the Paint icon at the top of the color palette.

The PICT editor opens and the paint tools palette appears.

- 3 Using the marquee tool, select the area you want to crop.
- 4 Press Return to erase the selected area.

You can use other tools for more sophisticated cropping. See “Using the Color Paint Tools Palette” on page 58.

## Dithering

Sometimes the PICT image you want to use has more colors than your computer system can display. For example, a PICT image might have been scanned using millions of colors, but your monitor may display only 256 colors.

In such a case, your computer uses the closest available colors from the system palette to display the image. Sometimes this causes banding in large swatches of gradually changing color.

HyperTalk's `Picture` command overcomes these problems by using dithering, a process that helps the computer approximate colors. Dithering creates patterns made up of available colors to simulate missing colors. For details on the `Picture` command, see Chapter 3 and the *HyperCard Script Language Guide*.

If your monitor can display more than 256 colors, you can use HyperTalk to change the color depth that the Color Tools use. In the stack's `openStack` handler, replace the line `addColor install` with `addColor install, 16` for thousands of colors, or `addColor install, 24` for millions of colors. Note that using a greater color depth requires more memory and slows stack performance.

## Preparing to Create a New PICT Image

To create a new PICT image, follow these steps:

- 1 Choose Open Coloring Tools from the Color menu.

The color editor opens with the color palette on the display.



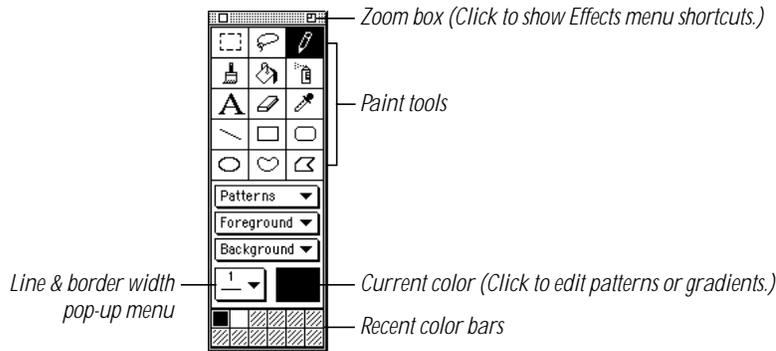
- 2 Choose Create New Picture from the Items menu.

Or double-click the Paint icon at the top of the color palette.

The PICT editor opens, and the color paint tools palette replaces the color palette. A new window appears in which you can draw a new PICT image.

## Using the Color Paint Tools Palette

You use the color paint tools palette to create and edit PICT images.



To add color to an image, follow these steps:

- 1 Click the PICT window whose image you want to edit.
- 2 Choose a pattern from the Patterns pop-up menu.
- 3 Choose a foreground color from the Foreground pop-up menu.

Or click a color on the Recent Color bars.

The foreground color is for the part of the pattern that shows as black in the Patterns menu. Thin straight lines, thin borders around shapes, and plain text also appear in the foreground color.

- 4 Choose a background color from the Background pop-up menu.

Hold down the Option key and click a color on the Recent Color bars.

The background color is for the part of the pattern that shows as white in the Patterns menu. The inside parts of wide lines, wide borders, and hollow text also appear in the background color.

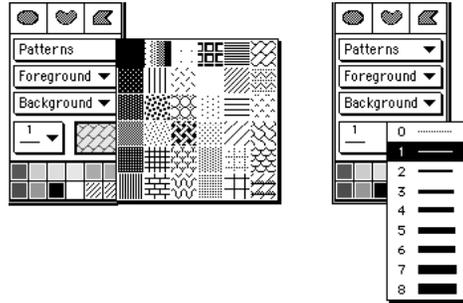
- 5 Choose a paint tool and edit the image.

The color sample (called the *current color*) near the bottom of the palette is a composite of the current pattern, foreground color, and background color. Whatever you paint uses this composite.



## Choosing and Editing Patterns

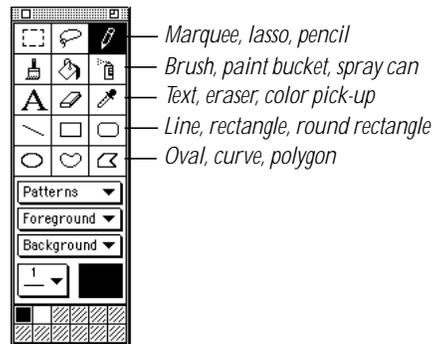
You use the Patterns pop-up menu to choose a pattern. The black pixels in a pattern will appear in the current foreground color; the white pixels will appear in the current background color.



- When you choose a pattern from the Patterns pop-up menu, the pattern appears in the current color box.
- To edit a pattern, choose Edit Patterns from the Options menu, click the pattern, and then edit it pixel by pixel.

## About the Tools

To select a tool, click it.



The shape of the pointer may change depending on the tool you select. Most of the color paint tools work like HyperCard's black-and-white paint tools.

### Marquee tool

Use the marquee tool to select a rectangular area of an image.

- To select an area, drag diagonally across it.
- To select the entire image, double-click the marquee tool icon in the paint palette.

Or choose Select All from the Edit menu.

- To shrink to the selection, hold down the Option key as you drag.  
When you release the mouse button, the selection rectangle shrinks around the image, eliminating the extra white space.
- To move a selection, put the pointer inside a selection and press the mouse button until the pointer changes to an arrow; then drag.
- To move a selection straight up or down or left or right, press the Shift key as you drag.
- To make a copy of the selection, press the Option key as you drag.
- To make multiple copies of a selection, press ⌘-Option as you drag.
- To move an image one pixel at a time, select it; then press an arrow key.
- To move an image five pixels at a time, select it; then hold down the Shift key as you press an arrow key.

#### **Lasso tool**

Use the lasso tool to select a specific part of an image too small for the marquee tool, a non-rectangular part of an image, or an image that is between other graphics.

- To use the lasso tool, drag around the area you want to select.  
The line that trails the lasso's tip shows what you are selecting. When you release the mouse button, the lasso shrinks around the image and selects it.
- To lasso an area without shrinking around the image, press the Option key as you drag.
- To lasso the whole image without selecting white space, double-click the lasso icon.
- To move a selection, position the lasso tool's tip within the selection until the pointer changes to an arrow; then drag.
- To move a selection straight up or down or directly left or right, press the Shift key as you drag.
- To make a copy of the selection, press the Option key as you drag.
- To make multiple copies of the selection, press ⌘-Option as you drag.
- To move an image one pixel at a time, select it; then press an arrow key.
- To move an image five pixels at a time, select it; then hold down the Shift key as you press an arrow key.

#### **Pencil tool**

Use the pencil tool to draw thin free-form lines.

The pencil draws in the selected color. If you click a pixel of the selected color and then draw, the pencil draws in the background color.

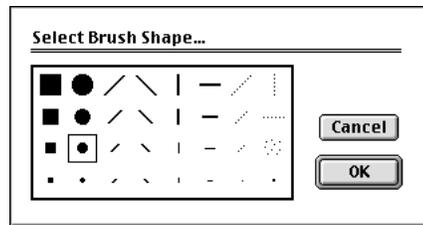
- To draw straight lines in a vertical or horizontal direction, press the Shift key as you drag.

- To zoom in to a higher magnification, double-click the pencil icon. Or press **⌘** and click the area that you want magnified.
- To zoom out to standard magnification, press the **⌘** and Shift keys and click.
- To move the entire work area, press the Option key as you drag.

### **Brush tool**

Use the brush tool to paint with the current pattern and brush shape.

- To use a new brush shape, choose Select Brush from the Options menu.  
Or double-click the brush in the palette.  
When the dialog box appears, click the shape you want to use.



Using different brush shapes and patterns creates different effects.

- To paint straight lines in a vertical or horizontal direction, press the Shift key as you drag.
- To use the brush tool as an eraser, press the **⌘** key as you drag.

### **Paint Bucket tool**

Use the paint bucket tool to fill solid areas of an image or hollow parts of outlined and shadowed text with the selected pattern and color.

**Important** Zoom in to check for gaps before using the paint bucket tool. If there is a gap or space in the outline of an area, paint spills out and fills the surrounding area. (You can choose Undo from the Edit menu to correct the mistake.) See “Zooming” on page 69 for instructions.

- To fill an area with the currently selected pattern and color, place the tip of the bucket in the area and click.

### **Spray Can tool**

Use the spray can tool to spray color onto an area.

- To set the width (Aperture) and density (Pressure) of the spray, double-click the spray can icon.



### **Text tool**

Use the text tool to add text to the image.

You can change the text font, size, and style by making choices from the Text menu immediately after typing the text and before clicking anywhere on the screen.

- To select the text you just typed, click the marquee or lasso tool.
- To select a font, choose Font from the Text menu and choose a font from the submenu. The current font is checked.
- To select a size, choose Size from the Text menu and choose a size from the submenu. The current size is checked. The best sizes for the current font are outlined.
- To see the Text Size dialog box, double-click the text tool icon.
- To select unlisted sizes, choose Size from the Text menu and drag to Other. When the dialog box appears, click the up or down arrow until you see the size you want to use in the text box (or type the size into the text box).
- To select a style for your text, choose Style from the Text menu and drag to the style you want to use. You can choose multiple font styles for the same text.
- To justify your text, choose Style from the Text menu and drag to the justification you want to use.

### **Eraser tool**

Use the eraser tool to erase part or all of the image.

- To erase part of the image, drag over the part you want to remove.
- To erase the entire image, double-click the eraser tool icon. If you make a mistake, choose Undo from the Edit menu.

### **Color Pick-Up tool**

Use the color pick-up tool to select a color from part of the image.

- To pick up any color on the screen, position the color pick-up tool over the color and click.

The current color box shows a solid foreground pattern and the color you click.

- To change all instances of a color to the current color, press the Option key and click the color with the color pick-up tool.
- To change all instances of a color to the current gradient, hold down the  $\mathbb{A}$  and Option keys, then click the color with the color pick-up tool.

See “Creating a Gradient” on page 70 for information on gradients.

- To toggle between the current tool and the color pick-up tool, press the Tab key.

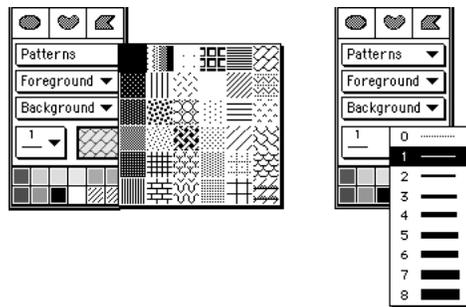
### Line tool

Use the line tool to draw straight lines.

- To draw lines in the selected pattern, press the Option key as you drag.
- To draw straight horizontal or vertical lines, press the Shift key as you drag.
- To change the line thickness, use the Line Width pop-up menu on the palette.

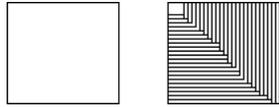
### Using the Closed Shape tools

To create any closed shape, click a shape tool; then select a pattern and a border size from the palette’s pop-up menus.



- To draw centered on a given point, choose Draw Centered from the Options menu before you drag.
- To draw borderless shapes, choose Draw Filled from the Options menu and press the Option key as you drag.  
Or select 0 from the Line Width pop-up menu.

- To draw multiple shapes, choose Draw Multiple from the Options menu before you drag.



**Draw  
Multiple off**

**Draw  
Multiple on**

- To create squares or circles, press the Shift key as you drag.
- To draw rectangles, use the rectangle tool.
- To draw rectangles with rounded corners, use the rounded rectangle tool.
- To draw oval shapes, use the oval tool.
- To draw a curved shape, use the curve tool. Position the pointer where you want to start; then drag to draw a shape. If you don't connect the start and end points of the shape, when you release the mouse button, the tool will connect them for you.
- To create polygons with irregular sides:
  - 1 Position the pointer where you want to begin; then click.
  - 2 Without pressing the mouse button, move the pointer to a second point and click, then to the third point and click, and so on until you're finished.
  - 3 To complete the polygon, double-click.
  - 4 You can now move the mouse without drawing any more lines.

## Setting Effects

The PICT editor provides a wide range of paint effects.

- To apply an effect to an entire PICT image, choose the effect you want to use from the Options menu.

The effect will change the PICT image in the active PICT window.

- To apply an effect to part of a PICT image, select the part you want to enhance, then choose an effect from the Options menu.

The effect will change only the selected part of the PICT image in the active PICT window.

The wording of a menu command may change to reflect whether that command will enhance the entire PICT image or just the part you have selected.

**Fill**

- To fill the image or the selection with the current pattern and colors, choose Fill from the Effects menu.

Or use the paint bucket tool.

**Invert**

- To change the color of pixels to colors on the opposite side of the color spectrum, choose Invert from the Effects menu.

**Tint**

- To tint the image by adding the foreground color to pixels, choose Tint from the Effects menu; then choose Toward Foreground from the submenu that appears.
- To tint the image by adding the background color to pixels, choose Tint from the Effects menu; then choose Toward Background from the submenu that appears.

**Anti-Alias**

- To remove jagged edges and outlines from the image, choose Anti-Alias from the Effects menu.

Anti-aliasing smooths the edges of the image by creating grayish intermediate pixels so that the edges become slightly blurred.

**Trace Edges**

- To outline edges, choose Trace Edges from the Effects menu.

Or press ⌘-E.

Repeated tracing adds more outlines to the edges.

**Rotate**

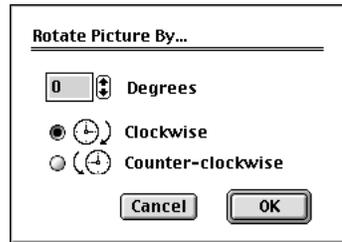
When you rotate an image, parts of it may be lost if they extend beyond the window. You can prevent this problem by expanding the size of the window before rotating the image.

- To turn the image 90 degrees right, choose Rotate from the Effects menu; then choose Right from the submenu that appears.
- To turn the image 90 degrees left, choose Rotate from the Effects menu; then choose Left from the submenu that appears.
- To turn a selected image by hand to any degree of rotation, choose Rotate from the Effects menu; then choose Free from the submenu that appears.

Drag the image by the handles that appear.

- To turn the image by a specific number of degrees, choose Rotate from the Effects menu; then choose By Degree from the submenu that appears.

When a dialog box appears, type the number of degrees of rotation that you want, click Clockwise or Counter-clockwise, and click OK.



- To turn the image by an additional amount after it has already been rotated, deselect it, then select and rotate it again.

### Scale

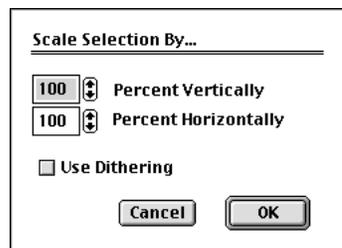
When you scale an image, parts of the image that extend beyond the window may be lost. You can prevent this problem by expanding the size of the window before scaling the image.

- To manually scale a selected image, choose Scale in the Effects menu; then choose Free from the submenu that appears.

Drag the image by the handles that appear.

- To scale the image by a specific percentage, choose Scale from the Effects menu; then choose By Percent from the submenu that appears.

In the dialog box, type the percent scaling that you want.



To dither the image when it scales, click Use Dithering in the Scale dialog box. (See “Dithering” on page 56.)

To scale the image proportionately, use the same number in both boxes.

### Flip

- To flip the image vertically about its center line, choose Flip from the Effects menu; then choose Vertically from the submenu that appears.
- To flip the image horizontally about its center line, choose Flip from the Effects menu; then choose Horizontally from the submenu that appears.

### Opaque

- To make the image opaque so that nothing shows through white parts, choose Opaque from the Effects menu.

### Transparent

- To make the image transparent so that anything beneath the white parts shows through, choose Transparent from the Effects menu.

### Draw Filled

- To fill shapes with the current pattern and colors as you draw them, choose Draw Filled from the Options menu.

The shape tools appear filled when this command is in effect.



**Draw Filled off**

**Draw Filled on**

### Draw Centered

- To draw a shape starting at the shape's center point, choose Draw Centered from the Options menu.

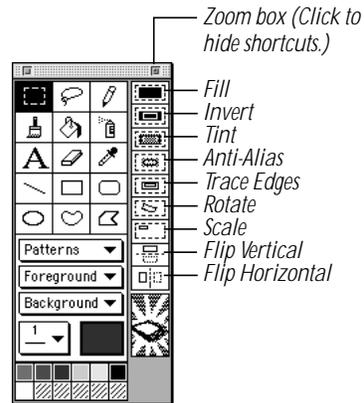
### Draw Multiple

- To draw multiple images as you drag a tool, choose Draw Multiple from the Options menu.

This command affects the line, rectangle, rounded rectangle, and oval tools.

## Paint Effects Shortcuts

When you click the Zoom box on the color paint tools palette, the palette expands to show a new group of icons. Click an icon to perform a paint effects shortcut.



## Importing and Exporting Images

You can import and export PICT images with the PICT editor.

### Importing an Image

To import an image into the PICT editor, follow these steps:

- 1 Choose Import Graphics from the File menu.  
A dialog box appears.
- 2 Click the types of files you want to import.
- 3 Locate the graphic file and click OK.

The imported image replaces everything in the active window.

Once an image has been imported, you can edit it.

### Exporting an Image

You can export any image from the PICT editor as a PICT file. To export an image, follow these steps:

- 1 Choose Export Graphics from the File menu.  
A dialog box appears.
- 2 Type a name for the new PICT file.
- 3 Move to the location where you want to store the file and click Save.

## **Saving a PICT Image**

The images you create or edit in the PICT editor are automatically saved as resources in the current stack when you close the editor.

- To save the changes you've made without closing the PICT editor, choose Save from the File menu.
- To revert to the most recently stored version of the image, choose Revert from the File menu.

## **Special Features of the PICT Editor**

The PICT editor has several features that give you more control over editing your images.

### **Editing Colors**

If your monitor's color depth is set to display thousands of colors or more, you can use the color picker to change colors in the PICT editor's palette.

To make changes to an existing color, follow these steps:

- 1** Choose Edit Colors from the Options menu.  
A dialog box with a color palette appears.
- 2** Click the color that you want to change.  
The color is surrounded by a thin black line to indicate that it's selected.
- 3** Click the Edit button.  
Or double-click the color.  
The color picker dialog box appears.
- 4** Click a color picking system on the left, then choose a new color on the right.  
The color picking systems available depend on the color picking software installed on your computer.
- 5** Click OK when you're finished.

### **Zooming**

You use zooming to magnify a section of an image for close-up work. You can magnify an image two, four, or eight times.

You can press a number key (1, 2, 3, or 4) to set the magnification.

- To zoom in one level of magnification, choose Zoom In from the Options menu.  
Or press ⌘J, or hold down the ⌘ key and click the image with the pencil tool.

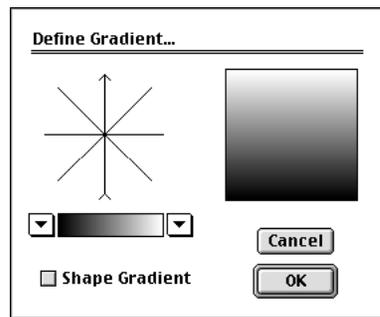
- To zoom out one level of magnification, choose Zoom Out from the Options menu.  
Or press  $\mathbb{A}$ -L, or  $\mathbb{A}$ -Shift and click the image with the pencil tool.

### Creating a Gradient

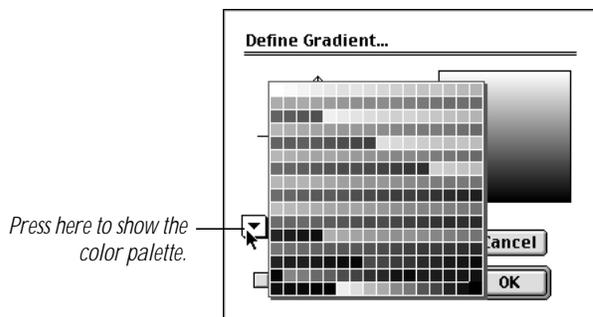
To create a gradient, a combination of two colors that gradually blend as they fill a shape, follow these steps:

- 1 Choose Edit Gradient from the Options menu.

A dialog box appears.



- 2 Press the arrow on the left of the gradient bar and choose a color from the pop-up palette.



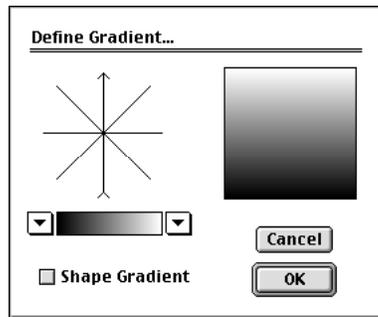
The color you choose becomes the color for the gradient's left side.

- 3 Press the arrow to the right of the gradient bar and choose a second color from the pop-up palette.

The color you choose becomes the color for the gradient's right side.

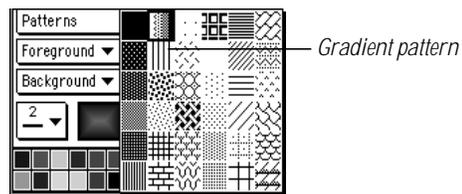
- 4 Click one of the eight lines above the gradient bar, or click the center of the lines once or twice.

A directional arrow appears to indicate the angle for the gradient. Click the center to create a radial gradient, one that radiates in or out from the center.



Click the Shape Gradient box if you want the gradient to adjust to the shape of the object it fills.

- 5 When you've finished editing the gradient, click OK.
  - To use a gradient to fill shapes, select the second pattern on the top row in the Patterns pop-up menu.



The edited gradient appears in the current color box.

### Closing the PICT Editor

- To close the PICT editor, close all paint windows, or choose Close Paint Tools from the File menu.

You return to the standard HyperCard environment.



## Scripting for Color

Anything you can do using the commands in the color editor you can also do by using scripts. This chapter describes the scripting functions that the Color Tools add to HyperTalk, HyperCard's scripting language.

See Chapter 3 of the *HyperCard Reference Manual* for a tutorial introduction to HyperTalk. See the *HyperCard Script Language Guide* for details about scripting and to learn about the rest of the commands, properties, and functions in HyperTalk.

### Syntax Terms and Conventions Used in This Chapter

The syntax descriptions in this chapter use the following conventions:

- Words and punctuation in non-italic monospaced type `like this` should be typed exactly as they appear.
- Words in italic type *like this* are parameter placeholders. You need to replace these words with specific instances. The description for each function makes clear what instances are appropriate.
- Curly braces { } enclose a pair of placeholders from which you must choose. The choices are separated by a vertical bar |.
- Square brackets [ ] enclose optional parameters.
- *Bevel* is a number between 0 and 6, inclusive, representing the width in pixels of a bevel on a button, field, or rectangle.
- *Color* is a set of three RGB numbers that together describe a single color.
- *Index* is the color layer that an overlay occupies (if getting the value), or the layer that you want an overlay to follow (if setting the value). New overlays are drawn at the front color layer. Passing a value of 0 (zero) creates the color overlay at the back; passing a value of -1 creates the overlay at the front. Each color layer holds a single overlay.
- *Pict* is the name of a PICT image, including images created with the color paint tools.
- *Pt* is a pair of numbers, separated by a comma, describing a point relative to the upper-left corner of the card.

- *Rect* is a set of four numbers, separated by commas, describing a rectangle relative to the upper-left corner of the card. The numbers represent the distance in pixels between the
  - left edge of the card and the rectangle's left edge
  - top of the card and the rectangle's top edge
  - left edge of the card and the rectangle's right edge
  - top of the card and the rectangle's bottom edge
- *Effect* is any one of the following (similar effects are grouped together):
  - `fromLeft,fromRight,fromTop,fromBottom,fromTopLeft,fromTopRight,fromBottomLeft,fromBottomRight`
  - `dissolve`
  - `irisOpen,irisClose`
  - `checkerBoardOpen,checkerBoardClose,circleCheckerOpen,circleCheckerClose`
  - `barnDoorOpen,barnDoorClose`
  - `combVertical,combHorizontal`
  - `rectOpen,rectClose`
  - `venetianBlindsHorizontal,venetianBlindsVertical`
  - `rakeHorizOpen,rakeHorizClose,rakeVertOpen,rakeVertClose`
- A color overlay is an item created or placed through a command in the color editor or by any `addColor` command. Such items include
  - color on a button or field
  - a color rectangle
  - any PICT image displayed on the screen (except those displayed by HyperTalk's Picture XCMD), including those created with the color paint tools

### **The addColor XCMD**

The main external command (XCMD) that makes the Color Tools work is `addColor`. It's installed into every stack to which you add color. You use the `addColor` XCMD to create scripts that add color to a stack while it's running.

### **addColor Structure**

Most of the color commands are actually parameters for `addColor`. The general structure of an `addColor` command is as follows:

```
addColor function, layer, parameter list
```

where *function* is any `addColor` function, *layer* is either `cd` (for card) or `bg` (for background), and *parameter list* is one or more parameters associated with a specific function.

The parameters must be entered in exactly the order shown. There are no default positional values. If you don't set all non-optional parameters, the script fails with the error returned in the HyperCard function `the result`.

If a single parameter has several items associated with it and you're using a literal for the items list (as opposed to a single variable), the items list must be enclosed within quotation marks with items separated by commas. For example, in the syntax line

```
addColor changeObjectColor, {cd|bg}, index, color
```

*color* takes three RGB numbers—one each for red, green, and blue:

```
addColor changeObjectColor, cd, 3, "65535, 32767, 16384"
```

The code can also read as follows:

```
put "65535, 32767, 16384" into theColor
addColor changeObjectColor, cd, 3, theColor
```

*Check the result:* `addColor` returns values as well as errors in the HyperCard function `the result`. Because all Color Tools errors are silent, `the result` is a particularly valuable debugging aid.

## AC\_RemoteInstall

```
AC_RemoteInstall [logicalExpr]
```

---

```
AC_RemoteInstall
```

```
AC_RemoteInstall true
```

---

*logicalExpr* resolves to either `true` or `false`.

This command installs the `addColor` XCMD under script control. Using it has the same effect as choosing Open Coloring Tools from the Color menu in a stack for the first time.

You use the optional parameter *logicalExpr* to determine if the scripts of the current stack should be modified. The preset value is `true`. If you set this parameter to `false`, the `addColor` resources are installed in the new stack but the stack's scripts remain unaffected.

The following handler (taken from the Color Tools' stack script) installs the `addColor` XCMD into a new stack and colors a button in the stack red.

```
on SetUpNewStack
  AC_RemoteInstall
  AddColor "addButton", "cd", 1, "65535,0,0", 6, -1
end SetUpNewStack
```

## **addButton**

```
addColor addButton, {cd|bg}, ID, color, bevel, index
```

```
addColor addButton, cd, 1, "20000,40000,30000", 3, -1
addColor addButton, bg, ID of bg btn "Choices", -
  theColor, 2, 3
```

This command adds a color overlay of the RGB color *color* with beveling level *bevel* to the button whose ID is *ID* on the current card or background. The overlay is initially placed in the frontmost color layer, and moves to color layer *index* at the next call to `colorCard`.

The following handlers create an orange oval button with a bevel of 3.

```
on newColorButton
  makeNewButton
  addColor addButton, cd, ID of last button, -
    "65535,32767,16384", 3, 1
  addColor colorCard -- puts overlay at proper level
end newColorButton

on makeNewButton
  lock screen
  set userLevel to 4 -- to make Objects menu available
  doMenu "New Button"
  set showName of last card button to false
  set the style of the last card button to oval
  set height of last card button to 100
  set width of last card button to 100
  choose Browse tool -- to deselect the new button
  unlock screen
end makeNewButton

Also see: colorButton
  getButtonIndex
  removeButton
```

## addField

```
addColor addField, {cd|bg}, ID, color, bevel, index
```

```
addColor addField, cd, 1, "20000,40000,30000", 3, -1
```

```
addColor addField, bg, ID of fld "Names", -1  
theColor, 2, 3
```

This command adds a color overlay of the RGB color *color* with beveling level *bevel* to the card or background field whose ID is *ID*. The overlay is initially placed in the frontmost color layer, and it moves to color layer *index* at the next call to `colorCard`.

The following handlers create a yellow opaque background field with a bevel of 3. (The `wideMargins` property is set to true to allow extra margin room for the bevel.)

```
on newColorField  
  makeNewField  
  put "65535,65535,39321" into yellow  
  addColor addField, bg, ID of last field, yellow, 3, 1  
  addColor colorCard -- puts overlay at proper level  
end newColorField  
  
on makeNewField  
  lock screen  
  doMenu "Background"  
  set userLevel to 4 -- to make Objects menu available  
  doMenu "New Field"  
  set wideMargins of last field to true  
  doMenu "Background"  
  choose Browse tool -- to deselect the new field  
  unlock screen  
end makeNewField
```

*Also see:* `colorField`  
`getFieldIndex`  
`removeField`

## **addPict**

```
addColor addPict, {cd|bg}, pict, {pt|rect}, {t|o}, index
```

```
addColor addPict,cd,"new one","0,0",t,3 addColor  
addPict,bg,"test pict","20,20,100,100",o,-1
```

This command locates the PICT resource named *PICT* in the current stack and displays it on the card or background. When you specify a point (*pt*), the PICT image appears full-sized with its upper-left corner at that point; when you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

The PICT image is initially placed in the frontmost color layer, and it moves to color layer *index* at the next call to `colorCard`.

When you set the opacity parameter to *t*, the white portions of the PICT image are transparent and anything below the PICT image shows through; when you set it to *o*, the entire PICT image is opaque.

The following handler asks the user for the name of a PICT image, and then places the PICT image on the screen full-sized with its upper-left corner at the pointer position.

```
on pictResource  
    ask file "What picture should I use?" of type PICT  
    addColor addPict,cd,it,the mouseLoc,o,0 -- in the back  
    addColor colorCard -- puts overlay at proper level  
end pictResource
```

*Also see:* `addPictFile`  
`colorPict`  
`colorPictFile`  
`getPictName`  
`removeObject`

## addPictFile

```
addColor addPictFile,{cd|bg}, pictFile, {pt|rect}, ~  
  {t|o}, index
```

```
addColor addPictFile,cd,"house","0,0",t,3  
addColor addPictFile,bg,team,"20,20,100,100",o,-1
```

This command locates the named PICT file and displays it on the card or background. (To display a PICT resource, use `addPict`.) When you specify a point (*pt*), the PICT image appears full-sized with its upper-left corner at that point. When you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

The PICT image is initially placed in the frontmost color layer, and it moves to color layer *index* at the next call to `colorCard`.

When you set the opacity parameter to *t*, the white portions of the PICT image are transparent and anything below the PICT image shows through; when you set it to *o*, the entire PICT image is opaque.

**Important** Your PICT files (or aliases that have the same names as the original files) must be in the same folder as HyperCard, Home, or your stack. Do not rename PICT files that you are displaying; HyperCard looks for PICT files by name.

The following handler asks the user for the name of a PICT image, and then displays the PICT image full-sized with its upper-left corner at the pointer position.

```
on pictureOnDisk  
  ask "What picture should I use?"  
  addColor addPictFile,cd,it,the mouseLoc,o,0  
  addColor colorCard -- puts overlay at proper level  
end pictureOnDisk
```

*Also see:* `addPict`  
 `colorPict`  
 `colorPictFile`  
 `getPictName`  
 `removeObject`

## **addRect**

```
addColor addRect, {cd|bg}, rect, color, bevel, index
```

```
addColor addRect, cd, "0,0,100,100", "30000,40000,50000", 3,0
```

```
addColor addRect, bg, myRectangle, myColor, 2, -1
```

This command creates a color rectangle in the RGB color *color* with beveling level *bevel* in the current card or background. The rectangle appears in a size and position specified by *rect*. It is initially placed in the frontmost color layer, and it moves to color layer *index* at the next call to `colorCard`.

The rectangle is a pure color object.

The following handler adds a red rectangle to color the current background.

```
on fillBack
  put the rect of this cd into theRect
  put "65535,0,0" into theColor
  put 4 into theBevel
  put 0 into theLayer -- 0 for backmost

  addColor addRect, bg, theRect, theColor, theBevel, theLayer
  if the result is not empty then
    answer "Error adding a Rectangle to Database:"-
    & return & the result
    exit fillBack
  end if
end fillBack
```

*Also see:* `colorRect`  
`removeObject`

### changeObjectBevel

```
addColor changeObjectBevel, {cd|bg}, index, bevel
```

```
addColor changeObjectBevel, cd, theIndex, newBevel
```

```
addColor changeObjectBevel, bg, 2, 5
```

This command changes the beveling level of the color overlay at color layer *index* on the current card or background to beveling level *bevel*.

If *bevel* is 0, any beveling already assigned to the overlay is removed.

The change to the overlay won't be visible until you make a call to `colorCard`.

**Important** This command will not create an overlay where none exists, nor will it add a bevel to a transparent button or field.

The bevel is a quality of the color overlay and not of its associated button or field.

The following handler changes the bevel of the card object under the pointer.

```
on changeBevel
  put the mouseLoc into thePoint
  addColor getObjectClicked, cd, thePoint
  put item 1 of the result into theIndex
  if theIndex is -1 then
    answer "There's no bevel to change."
    exit changeBevel
  end if
  addColor getObjectBevel, cd, theIndex
  put the result into oldBevel
  repeat
    put random(7) - 1 into newBevel
    if oldBevel <> newBevel then exit repeat
  end repeat
  addColor changeObjectBevel, cd, theIndex, newBevel
  addColor colorCard
end changeBevel
```

*Also see:* getObjectBevel

### **changeObjectBounds**

```
addColor changeObjectBounds, {cd|bg}, index, rectangle
```

---

```
addColor changeObjectBounds, cd, 4, "10,10,100,100"  
addColor changeObjectBounds, bg, theIndex, newRectangle
```

---

This command changes the rectangle of the PICT image or color rectangle on the current card or background in the specified color layer (*index*). It has no effect on buttons or fields or on their associated color overlays.

Changes to the rectangle won't be visible until you make a call to `colorCard`.

The following handler moves the card object under the pointer one-half inch down and to the right.

```
on changeBounds  
  put the mouseLoc into thePoint  
  addColor getObjectClicked, cd, thePoint  
  
  get the result  
  
  put item 1 of it into theIndex  
  put item 2 of it into theType  
  if theType < 3 then --Make sure object is right type  
    answer "There's no rectangle or PICT image here."  
    exit changeBounds  
  end if  
  
  addColor getObjectBounds, cd, theIndex  
  get the result  
  repeat with i = 1 to 4  
    add 36 to item i of it -- 36 pixels = 1/2 inch  
  end repeat  
  
  addColor changeObjectBounds, cd, theIndex, it--Set new rect  
  addColor colorCard  
end changeBounds
```

*Also see:* getObjectBounds

### **changeObjectColor**

```
addColor changeObjectColor, {cd|bg}, index, color
```

```
addColor changeObjectColor, cd, 4, "30000,40000,60000"
```

```
addColor changeObjectColor, bg, theIndex, newColor
```

This command changes the color of the overlay in color layer *index* on the current card or background.

**Important** This command, which applies only to buttons, fields, and rectangles, will not create an overlay where none exists.

The following handler assigns a random color to the object under the pointer.

```
on changeColor
  put the mouseLoc into thePoint
  addColor getObjectClicked, cd, thePoint
  put item 1 of the result into theIndex

  repeat with theItem = 1 to 3
    put random(65535) into item theItem of theColor
  end repeat

  addColor changeObjectColor, cd, theIndex, theColor
  addColor colorCard
end changeColor
```

*Also see:* getObjectColor

### **changeObjectTransparency**

```
addColor changeObjectTransparency, {cd|bg}, index, {t|o}
```

```
addColor changeObjectTransparency, cd, 4, t
```

```
addColor changeObjectTransparency, bg, theIndex, o
```

This command sets the opacity of the PICT image in the designated color layer on the current card or background to either opaque (o) or transparent (t).

When a PICT image has its opacity parameter set to t, the white portions of the PICT image are transparent and any color below the PICT image shows through. When the opacity parameter is set to o, the entire PICT image is opaque.

The following handler toggles the transparency of the PICT image under the pointer.

```
on changeTransparency
  put the mouseLoc into thePoint
  addColor getObjectClicked, cd, thePoint
  get the result
  if item 2 of it < 4 then
    answer "The pointer isn't over a PICT image."
    exit changeTransparency
  end if

  put item 1 of it into theIndex
  addColor getObjectColor, cd, theIndex
  get the result
  if item 4 of it is "t"
    then put "o" into opacity
  else put "t" into opacity

  addColor changeObjectTransparency, cd, theIndex, opacity
  addColor colorCard
end changeTransparency
```

## colorBackground

```
addColor colorBackground [ , effect [ , duration ] ]
```

```
addColor colorBackground
```

```
addColor colorBackground, dissolve, 60
```

This command turns off color in the card level and adds color to the background with an optional transitional effect.

When you use a transition effect, you can also specify the duration of the transition in sixtieths of a second. The default duration is one second. Transition effects include the following (similar effects are grouped together):

- fromLeft, fromRight, fromTop, fromBottom, fromTopLeft, fromTopRight, fromBottomLeft, fromBottomRight
- dissolve
- irisOpen, irisClose
- checkerBoardOpen, checkerBoardClose, circleCheckerOpen, circleCheckerClose
- barnDoorOpen, barnDoorClose
- combVertical, combHorizontal
- rectOpen, rectClose
- venetianBlindsHorizontal, venetianBlindsVertical
- rakeHorizOpen, rakeHorizClose, rakeVertOpen, rakeVertClose

Use the command `addColor colorCard` to redraw the card colors.

The following handler, which assumes you have color items in both the card and background, turns the card color off and on several times.

```
on flashCard
  repeat 3
    addColor colorBackground, dissolve
    addColor colorCard, dissolve
  end repeat
end flashCard
```

*Also see:* `colorCard`  
`colorCardLayered`

## colorButton

```
addColor colorButton, {cd|bg}, ID, color, bevel
```

```
addColor colorButton, cd, 1, "20000,40000,30000", 3
```

```
addColor colorButton, bg, ID of bg btn "Choices", theColor, 2
```

This command adds a temporary color overlay in the RGB color *color* with beveling level *bevel* to the card or background button whose ID is *ID*. The overlay is placed in the frontmost color layer, and is removed at the next call to `colorCard` or `colorBackground`.

**Important** Even though the color overlay is drawn at the very front, you still need to specify whether the button is on the card or on the background.

The color you add to a button with this command is temporary. To add color to a button permanently, use the command `addColor addButton`.

The following handler colorizes three buttons with IDs 1, 2, and 3 on the card layer, maintains the color for two seconds, and then removes the color.

```
on tempButton
  repeat with buttonID = 1 to 3
    put "65535, 32767, 16384" into theColor
    put random (7)-1 into theBevel
    addColor colorButton, cd, buttonID, theColor, theBevel
  end repeat
  wait 120
  addColor colorCard
end tempButton
```

*Also see:* `addButton`  
`getButtonIndex`  
`removeButton`

## **colorCard**

```
addColor colorCard [ ,effect [ ,duration]]
```

```
addColor colorCard
```

```
addColor colorCard,dissolve,60
```

This command adds color to the current card, displaying all color overlays in their proper color layers.

You can also use this command to remove any PICT images or other color overlays temporarily added to the current card or background using `colorButton`, `colorField`, `colorRect`, `colorPict`, or `colorPictFile`.

You can specify a transition effect as the color appears on the card, and a duration for the transition in sixtieths of a second (ticks). Transition effects include the following (similar effects are grouped together):

- `fromLeft`, `fromRight`, `fromTop`, `fromBottom`, `fromTopLeft`, `fromTopRight`, `fromBottomLeft`, `fromBottomRight`
- `dissolve`
- `irisOpen`, `irisClose`
- `checkerBoardOpen`, `checkerBoardClose`, `circleCheckerOpen`, `circleCheckerClose`
- `barnDoorOpen`, `barnDoorClose`
- `combVertical`, `combHorizontal`
- `rectOpen`, `rectClose`
- `venetianBlindsHorizontal`, `venetianBlindsVertical`
- `rakeHorizOpen`, `rakeHorizClose`, `rakeVertOpen`, `rakeVertClose`

Because it is needed to redraw the screen properly, displaying colors where they belong, this command appears in nearly every color handler.

The `colorCard` command draws items on the screen in the following order:

- 1** All background objects, PICT images, and color rectangles
- 2** All card objects, PICT images, and color rectangles
- 3** HyperCard's black-and-white image

The following handler temporarily draws a blue rectangle on the screen, waits for the mouse button to be pressed, and then destroys the rectangle with a call to `colorCard`.

```
on waitForIt
  addColor colorRect, cd, "30,30,200,200", "0,0,65000", 6
  wait until the mouse is down
  addColor colorCard
end waitForIt
```

**Important** To avoid the sudden startling of color objects on the screen, lock the screen before you change cards; then go to the destination card and call `addColor colorCard`.

*Also see:* `colorBackground`  
`colorCardLayered`

## colorCardLayered

```
addColor colorCardLayered [ ,effect [ ,duration]]
```

```
addColor colorCardLayered  
addColor colorCardLayered, dissolve,60
```

This command adds color to the current card, displaying all color overlays in their proper color layers.

You can also use this command to remove any overlays temporarily added to the current card or background using `colorButton`, `colorField`, `colorRect`, `colorPict`, or `colorPictFile`.

You can specify a transition effect as the coloring appears on the card, and a duration for the transition in sixtieths of a second (ticks). Transition effects include the following (similar effects grouped together):

- `fromLeft`, `fromRight`, `fromTop`, `fromBottom`, `fromTopLeft`, `fromTopRight`, `fromBottomLeft`, `fromBottomRight`
- `dissolve`
- `irisOpen`, `irisClose`
- `checkerBoardOpen`, `checkerBoardClose`, `circleCheckerOpen`, `circleCheckerClose`
- `barnDoorOpen`, `barnDoorClose`
- `combVertical`, `combHorizontal`
- `rectOpen`, `rectClose`
- `venetianBlindsHorizontal`, `venetianBlindsVertical`
- `rakeHorizOpen`, `rakeHorizClose`, `rakeVertOpen`, `rakeVertClose`

The `colorCardLayered` command colors items in the following order, which is different from the order that `colorCard` uses:

- 1** Background PICT images and color rectangles
- 2** Card PICT images and colorized rectangles
- 3** Background button overlays and field overlays
- 4** Card button overlays and field overlays
- 5** HyperCard's black-and-white image

**Important** It takes twice as long to color a card using `colorCardLayered` as it does using `colorCard`.

The following handler shows how card layering works.

```
on showColoringSequence
  put "0,65000,0" into green
  put "65000,0,0" into red
  put "65000,65000,30000" into yellow
  put "0,0,65000" into blue

  put ID of last cd btn into lastCdBtnID
  put ID of last bg btn into lastBgBtnID

  addColor addButton, cd, lastCdBtnID, green, 3, -1
  addColor addButton, bg, lastBgBtnID, yellow, 3, -1
  addColor addRect, cd, "0,0,100,100", red, 5, -1
  addColor addRect, bg, "100,100,200,200", blue, 5, -1
  addColor remove addColor install
  addColor colorCardLayered, dissolve, 60
end showColoringSequence
```

*Also see:* colorCard  
colorBackground

## colorField

```
addColor colorField, {cd|bg}, ID, color, bevel
```

```
addColor colorField, cd, 1, "20000,40000,30000", 3
```

```
addColor colorField, bg, ID of fld "Animals", theColor, 2
```

This command adds a temporary color overlay in the RGB color *color* with beveling level *bevel* to the field whose ID is *ID* in the specified card or background domain. The overlay is placed in the frontmost color layer and is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

**Important** Even though the color overlay is drawn at the very front, you still need to specify whether the field is on the card or on the background.

The color you add to a field with this command is temporary. To add color to a field permanently, use the command `addColor addField`.

The following handler adds color to three background fields with IDs 1, 2, and 3, maintains the color for two seconds, and then removes it.

```
on tempField
  repeat with fieldID = 1 to 3
    put "65535, 0, 0" into theColor
    put random(7)-1 into theBevel
    addColor colorField, cd, fieldID, theColor, theBevel
  end repeat
  wait 120
  addColor colorCard
end tempField
```

*Also see:* `addField`  
`getFieldIndex`  
`removeField`

## colorPict

```
addColor colorPict, {cd|bg}, pict, {pt|rect}, {t|o}, -  
[ , effect [ , duration]]
```

```
addColor colorPict, cd, "new one", "0,0", t  
addColor colorPict, bg, "test pict", "20,20,100,100", o, -  
fromTopLeft
```

This command locates the PICT resource named *pict* in the current stack and temporarily displays it on the current card or background. When you specify a point (*pt*), the PICT image appears full-sized with its top left corner at that point. When you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

You can specify a transition effect as the PICT image appears on the screen, and a duration for the transition in sixtieths of a second (ticks). Transition effects include the following (similar effects are grouped together):

- fromLeft, fromRight, fromTop, fromBottom, fromTopLeft, fromTopRight, fromBottomLeft, fromBottomRight
- dissolve
- irisOpen, irisClose
- checkerBoardOpen, checkerBoardClose, circleCheckerOpen, circleCheckerClose
- barnDoorOpen, barnDoorClose
- combVertical, combHorizontal
- rectOpen, rectClose
- venetianBlindsHorizontal, venetianBlindsVertical
- rakeHorizOpen, rakeHorizClose, rakeVertOpen, rakeVertClose

The PICT image appears in the frontmost color layer, and is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

**Important** Even though the PICT image is drawn at the very front, you still need to specify whether the PICT image is on the card or on the background.

A PICT image you display with this command is temporary. If you want to display a PICT resource each time you open the card, use the command `addColor addPict`.

When you set the opacity parameter to `t`, the white portions of the PICT image become transparent and any color below the PICT image shows through. When you set the opacity parameter to `o`, the entire PICT image is opaque.

The following handler temporarily puts a PICT resource on the screen with its upper-left corner at the pointer. The PICT image is removed when the mouse button is pressed.

```
on tempPictResource
  ask "What picture should I use?"
  addColor colorPict,cd,it,the mouseLoc,o,irisOpen
  wait until the mouse is down
  addColor colorCard
end tempPictResource
```

*Also see:* addPict  
addPictFile  
colorPictFile  
getPictName  
removeObject

## colorPictFile

```
addColor colorPictFile, {cd|bg}, pict, {pt|rect}, {t|o}, -  
[ , effect [ , duration]]
```

```
addColor colorPictFile, cd, "new one", "0,0", t  
addColor colorPictFile, bg, "test pict", "20,20,100,100", o, -  
fromTopLeft
```

This command temporarily displays a copy of the PICT file named *pict*. (To temporarily display a PICT resource, use `colorPict`.) When you specify a point (*pt*), the PICT image appears full-sized with its upper-left corner at that point. When you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

You can specify a transition effect as the PICT image appears on the screen, and a duration for the transition in sixtieths of a second (ticks). Transition effects include the following (similar effects are grouped together):

- `fromLeft`, `fromRight`, `fromTop`, `fromBottom`, `fromTopLeft`, `fromTopRight`, `fromBottomLeft`, `fromBottomRight`
- `dissolve`
- `irisOpen`, `irisClose`
- `checkerBoardOpen`, `checkerBoardClose`, `circleCheckerOpen`, `circleCheckerClose`
- `barnDoorOpen`, `barnDoorClose`
- `combVertical`, `combHorizontal`
- `rectOpen`, `rectClose`
- `venetianBlindsHorizontal`, `venetianBlindsVertical`
- `rakeHorizOpen`, `rakeHorizClose`, `rakeVertOpen`, `rakeVertClose`

The PICT image appears in the frontmost color layer, and it is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

**Important** Even though the PICT image is drawn at the very front, you still need to specify whether the PICT image is on the card or on the background.

A PICT image you display with this command is temporary. If you want to display a PICT file each time you open the card, use the command `addColor addPictFile`.

When you set the opacity parameter to `t`, the white portions of the PICT image become transparent and any color below the PICT image shows through. When you set the opacity parameter to `o`, the entire PICT image is opaque.

The following handler temporarily displays a copy of a PICT file at the upper-left corner of the card. When the screen is redrawn, the PICT image will disappear.

```
on tempPictFile
  ask "What PICT file should I use?"
  put it into fileName
  addColor colorPictFile, cd, fileName, "0,0",t
end tempPictfile
```

*Also see:* addPict  
addPictFile  
colorPict  
getPictName  
removeObject

## colorRect

```
addColor colorRect, {cd|bg}, rect, color, bevel
```

```
addColor colorRect, cd, "0,0,100,100", "30000,40000,50000", 3
```

```
addColor colorRect, bg, myRectangle, myColor, 2
```

This command temporarily creates a color rectangle on the card or background in the RGB color *color* with beveling level *bevel*. The rectangle appears in a size and position specified by *rect*.

The rectangle is displayed in the frontmost color layer, and it is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

**Important** Even though the color overlay is drawn at the very front, you still need to specify whether the rectangle is on the card or on the background.

The rectangle that you display with this command is temporary. To display a color rectangle each time you open the card, use the command `addColor addRect`.

The rectangle is a pure color object.

The following handler paints color rectangles in random locations on the card until the mouse button is pressed; then the rectangles are all erased.

```
on randomTempRects
  repeat until the mouse is down
    repeat with color=1 to 3
      put random(65535) into item color of theColor
    end repeat

    put random(7)-1 into theBevel
    put random(100) into item 1 of theRect
    put random(100) into item 2 of theRect
    put item 3 of the rect of this card into btm
    put random(btm) + item 1 of theRect-
    into item 3 of theRect
    put item 4 of the rect of this card into rgt
    put random(rgt) + item 2 of theRect-
    into item 4 of theRect
```

*(continued)*

```
    AddColor colorRect,cd,theRect,theColor,theBevel
end repeat
--          Erase 'em all
    addColor colorCard
end randomTempRects

Also see: addRect
          removeObject
```

## compact

```
addColor compact, {cd|bg}
```

---

```
addColor compact, bg
```

```
addColor compact, cd
```

---

This command cleans up the Color Tools' internal database, removing information about deleted color buttons and fields.

To keep color information as up-to-date as possible, use this command whenever you delete a button or field to which you've added color.

You must issue this command separately for the card and the background, as demonstrated in the following handler:

```
on killButtons
  delete card button "Green"
  delete background button "Blue"
  addColor compact, bg
  addColor compact, cd
  addColor colorCard
end killButtons
```

The `compact` command doesn't remove the color of the deleted object from the screen. To do that, you must use `colorCard` or `colorBackground`, as appropriate.

The command `addColor compact` is not the same as `Compact Stack`. Using `addColor compact` compacts the color database but does not compact the stack. To compact the stack, use the standard HyperTalk command `doMenu Compact Stack` with the browse, field, or rectangle tool selected.

The following handler deletes a field from the background and properly cleans up after itself.

```
on killFieldCleanly
  delete background field "Redundant"
  addColor compact, bg
  addColor colorCard if the freeSize of this stack-
  >.05 * the size of this stack then
    get the tool -- what tool is in use?
    choose the browse tool
    doMenu "Compact Stack"
    choose it -- restore user's tool
  end if
end killFieldCleanly
```

## **depth**

```
addColor depth
```

---

```
addColor depth
```

---

This selector returns the current color depth of the color buffers.

The following handler, placed in the stack script, restricts the current stack to work only in 16 colors (thus reducing the amount of RAM the stack needs).

```
on openStack
  addColor install, 4
end openStack
```

*Also see:* `remove`  
`enable`

## disable

```
addColor disable
```

```
addColor disable
```

This command temporarily turns Color Tools off, preserving the off-screen memory buffers that hold color information. You use this command to prevent the Color Tools from interfering with an XCMD that you're about to run.

**Important** To turn Color Tools back on again, use the command `addColor enable`. Until you do so, all other `addColor` calls are ignored.

The following example turns off the Color Tools, runs an XCMD, and then turns the Color Tools back on:

```
on doAnotherXCMD
  addColor disable
  runMyZippyXCMD
  addColor enable
end doAnotherXCMD
```

The following handler attempts to color a button orange with the Color Tools turned off, then turns the tools on again to complete the job:

```
on stopTheShow
  addColor disable
  addColor addButton,cd,ID of btn "Books",-
  "65535,32767,16384",3,0
  addColor colorCard
  answer "Nothing works yet.."

  addColor enable
  addColor addButton,cd,ID of btn "Books",-
  "65535,32767,16384",3,0
  addColor colorCard
  answer "...but now it does."
end stopTheShow
```

*Also see:* `enable`  
`disableObject`  
`remove`

## **disableObject**

```
addColor disableObject, {cd|bg}, index
```

---

```
addColor disableObject, cd, 3
```

```
addColor disableObject, bg, theColorLayer
```

---

This command temporarily turns off an overlay's ability to show itself.

Only the color overlay is hidden. The button or field is still visible. To hide a button or field, use HyperTalk's `hide` command.

**Important** Before you disable an overlay, note its index value. While the overlay is disabled, you can't address it to learn its index, a value you'll need later to restore the overlay.

The following example hides the color overlay on the card under the pointer while you hold down the mouse button.

```
on hideWhenDown
    addColor getObjectClicked, cd, the mouseLoc
    put item 1 of the result into theIndex

    wait until the mouse is down
    addColor disableObject, cd, theIndex
    addColor colorCard

    wait until the mouse is up
    addColor enableObject, cd, theIndex
    addColor colorCard
end hideWhenDown
```

*Also see:* `enableObject`  
`disable`  
`remove`

## **enable**

```
addColor enable
```

---

```
addColor enable
```

---

This command turns the Color Tools on after they have been made inoperative by `disable`. The following example turns off the Color Tools, runs an XCMD, and then turns the Color Tools back on:

```
on doAnotherXCMD
    addColor disable
    runMyZippyXCMD
    addColor enable
end doAnotherXCMD
```

Choosing Open Coloring Tools from the Color menu also turns the Color Tools back on.

The following handler attempts to color a button red with the Color Tools turned off, then turns the tools on again to complete the job:

```
on stopTheShow
    addColor disable
    addColor addButton, cd, ID of btn "CDs", ↵
    "65535,0,0",3,0
    addColor colorCard
    answer "Nothing works yet.."

    addColor enable
    addColor addButton,cd, ID of btn "CDs", ↵
    "65535,0,0",3,0
    addColor colorCard
    answer "but now it does."
end stopTheShow
```

*Also see:* `disable`  
`enableObject`  
`install`

## **enableObject**

```
addColor enableObject, {cd|bg}, index
```

---

```
addColor enableObject, cd, 3
```

```
addColor enableObject, bg, theColorLayer
```

---

This command turns on a color object previously turned off by `disableObject`.

To display the object after you turn it on, use `addColor colorCard`.

The following handler hides the color of the card object under the pointer while the mouse button is down.

```
on showWhenUp
  addColor getObjectClicked, cd, the mouseLoc
  put item 1 of the result into theIndex

  wait until the mouse is down
  addColor disableObject, cd, theIndex
  addColor colorCard

  wait until the mouse is up
  addColor enableObject, cd, theIndex
  addColor colorCard
end showWhenUp
```

*Also see:* `enableObject`  
`disable`  
`install`

## **getBitsCall**

```
addColor getBitsCall
```

---

```
addColor getBitsCall
```

---

This command, designed for XCMD developers, returns the address of the addColor patch in HyperTalk's function `the result`. The address provides access to the Color Tools' color buffers for the current window.

The following handler displays the patch address in an answer dialog box.

```
on findPatchAddress
    addColor getBitsCall
    answer "Color buffers start at" && the result & "."
end findPatchAddress
```

### **getButtonIndex**

```
addColor getButtonIndex, {cd|bg}, /D
```

---

```
addColor getButtonIndex, cd, 3
```

```
addColor getButtonIndex, bg, ID of button "Belly"
```

---

This command returns the index value of the color overlay for the card or background button whose ID is *ID*.

The higher the number of the index value, the closer the color overlay is to the front of the color layer.

A value of  $-1$  means that the specified button has no color overlay.

The following handler reports the index value for the button under the pointer.

```
on locateButtonColor
  get the mouseLoc
  -- Is it a card button?
  repeat with here = number of card buttons down to 1
    if it is within the rect of card button here then
      get the Id of card button here

      addColor getButtonIndex, cd, it

      answer "The card button at the pointer is at index"-
        && the result && "."
      exit locateButtonColor
    end if
  end repeat

  --Is it a background button?
  repeat with here = number of background buttons down to 1
    if it is within the rect of bg button here then
      get the Id of background button here

      addColor getButtonIndex, bg, it

      answer "The button at the pointer is at index"-
        && the result && "."
      exit locateButtonColor
    end if
  end repeat
end locateButtonColor
```

*Also see:* addButton  
colorButton  
removeButton

### getFieldIndex

```
addColor getFieldIndex, {cd|bg}, ID
```

```
addColor getFieldIndex, cd, 3
```

```
addColor getFieldIndex, bg, ID of field "Elysium"
```

This command returns the index value of the color overlay for the card or background field whose ID is *ID*.

The higher the number of the index value, the closer the color overlay is to the front of the color layer.

A value of -1 means that the specified field has no color overlay.

The following handler reports the index value for the field under the pointer.

```
on locateFieldColor
  get the mouseLoc
  -- Is it a card field?
  repeat with here = number of card fields down to 1
    if it is within the rect of card field here then
      get the Id of card field here
      addColor getFieldIndex, cd, it
      answer "The card field at the pointer is at index"-
        && the result && "."
      exit to HyperCard
    end if
  end repeat
  -- Is it a background field?
  repeat with here = number of background fields down to 1
    if it is within the rect of bg field here then
      get the Id of background field here
      addColor getFieldIndex, bg, it
      answer "The field at the pointer is at index"-
        && the result && "."
      exit to HyperCard
    end if
  end repeatend
locateFieldColor
```

*Also see:* addField  
colorField  
removeField

## **getObjectBevel**

```
addColor getObjectBevel, {cd|bg}, index
```

---

```
addColor getObjectBevel, cd, 3
```

```
addColor getObjectBevel, bg, theIndex
```

---

This command returns the bevel width in pixels of the color overlay for the card or background button, field, or rectangle at color layer *index*. The bevel width is returned in the function the `result`.

Meaningful widths are 0 through 6. If the `result` is empty, you've provided an invalid index number.

The following handler reports the bevel of the card object under the pointer.

```
on getBevel
  addColor getObjectClicked, cd, the mouseLoc
  get item 1 of the result
  addColor getObjectBevel, cd, it
  answer "The bevel is" && the result && "."
end getBevel
```

*Also see:* `changeObjectBevel`

### **getObjectBounds**

```
addColor getObjectBounds, {cd|bg}, index
```

---

```
addColor getObjectBounds, cd, 3
```

```
addColor getObjectBounds, bg, theIndex
```

---

This command returns the bounding rectangle of the card or background color overlay at color layer *index*. The rectangle is returned as a comma-separated list of four numbers in the function the *result*.

If the *result* is empty, you've provided an invalid index number.

The following handler reports the rectangle of the card object under the pointer.

```
on getBounds
  addColor getObjectClicked, cd, the mouseLoc
  get the result
    put item 1 of it into theIndex
  addColor getObjectBounds, cd, theIndex
  answer "The rectangle of the object at the pointer is"-
    && the result & "."
end getBounds
```

*Also see:* changeObjectBounds

### **getObjectClicked**

```
addColor getObjectClicked, {cd|bg}, point[, type]
```

```
addColor getObjectClicked, cd, "100,300", buttonType
```

```
addColor getObjectClicked, bg, the mouseLoc
```

This command returns the index value and the type of card or background color overlay at point *point*. The information is returned in the function the *result*.

Optionally, you can specify in *type* the type of object that this command reports on. The type you specify must be one of the following:

- `buttonType`
- `fieldType`
- `rectType`
- `pictType`
- `pictFileType`

The first number returned indicates the index (that is, the color layer) in which the overlay exists. The second number, separated from the first by a comma, represents the type of overlay located, as follows:

<b>Returned value</b>	<b>Overlay type</b>
1	button
2	field
3	rectangle
4	PICT resource
5	PICT file

A result of `-1, -1` means that no color overlay of the specified type exists at the location you specified.

The following handler reports the type, location, and index of the card object under the pointer.

```
on getIndex
  put "button,field,rectangle,PICT resource,PICT file"~
  into theType
  put the mouseLoc into thePoint
  addColor getObjectClicked,cd,thePoint
  get the result
  if item 1 of it is -1 then
    answer "No card overlay there."
    exit getIndex
  end if
  put item (item 2 of it) of theType into theObject
  put item 1 of it into theIndex
  answer "The index of the" && theObject~
  && "at" && thePoint && "is" && theIndex & "."
end getIndex
```

### **getObjectColor**

```
addColor getObjectColor, {cd|bg}, index
```

---

```
addColor getObjectColor, cd, 3
```

```
addColor getObjectColor, bg, theIndex
```

---

This command returns a comma-separated list of either three or four values in the HyperTalk function the `result`. The first three values compose the RGB color of the card or background button, field, or rectangle located at color layer *index*. A fourth value is returned when the object is a PICT image; the value is the PICT image's opacity (t for transparent, o for opaque).

If the item is a PICT image, the color values are meaningless.

If the `result` is empty, you've provided an invalid index number.

The following handler reports the color or opacity of the item under the pointer.

```
on getColor
  addColor getObjectClicked, cd, the mousetloc
  addColor getObjectColor, cd, item 1 of the result
  get the result
  if it is empty
    then answer "The index number is invalid."
  else if the number of items in it = 3
    then answer "The object's RGB color is" && it & "."
  else answer "The PICT image's opacity value is" &
    && item 4 of it & "."
end getColor
```

*Also see:* `changeObjectColor`

## **getObjectType**

```
addColor getObjectType, {cd|bg}, index
```

---

```
addColor getObjectType, cd, 3
```

```
addColor getObjectType, bg, theIndex
```

---

This command returns the type of overlay located at color layer *index* in the HyperTalk function the `result`. The value returned is one of the following:

- `buttonType`
- `fieldType`
- `rectType`
- `pictType`
- `pictFileType`

If the `result` is empty, you've provided an invalid index number.

The following handler reports the type of overlay in the first five layers of color on the current card.

```
on identifyOverlay
  repeat with theIndex = 1 to 5
    addColor getObjectType, cd, theIndex
    put "Layer" && theIndex & ":" && the result &
      into line theIndex of typeList
  end repeat
  "in the first five color layers." &
  & return & return & typeList
end identifyOverlay
```

### **getPictName**

```
addColor getPictName, {cd|bg}, index
```

---

```
addColor getPictName, cd, 3
```

---

```
addColor getPictName, bg, theIndex
```

---

This command returns the name of the PICT resource or file located at color layer *index* in the HyperTalk function *the result*.

If *the result* is empty, you've provided an invalid index number.

The following handler reports the name of the PICT resource or file under the pointer.

```
on whatPICT
    addColor getObjectClicked, cd, the mousetloc
    addColor getPictName, cd, item 1 of the result
    get the result
    if it is empty then answer "Sorry - no PICT here."
    else answer "The PICT at the pointer is"~
        && it &". "
end whatPict
```

## **install**

```
addColor install [ ,colorDepth]
```

---

```
addColor install  
addColor install, 24
```

---

This command makes the Color Tools usable in the current stack, creating off-screen memory buffers to hold color information. It belongs in the stack script's `openStack` handler.

The Color Tools are installed with a preset color depth of 8, allowing 256 colors. To increase or decrease the color depth, use the optional *colorDepth* parameter.

This command, which must be called before any other color command, is automatically installed in the stack script when you first open the color editor.

If your monitor is set to a different color depth, you may not be able to use all of the colors in the palette.

## **moveBackward**

```
addColor moveBackward, {cd|bg}, index
```

---

```
addColor moveBackward, cd, 3
```

```
addColor moveBackward, bg, theIndex
```

---

This command moves the card or background color overlay at color position *index* one layer back.

The overlay's new position is returned in the HyperTalk function `the result`, and will be one number lower than the value you specify in *index*. If *index* holds the value 1, indicating that the overlay is already all the way in the back, the returned value will remain 1.

The overlay that holds the immediately lower position before this command is executed moves forward one layer. (In effect, the overlays switch layers.)

*Some index values may change:* Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following handler moves the overlay under the pointer one layer closer to the back.

```
on moveBack
    addColor getObjectClicked,cd,the mouseLoc
    put item 1 of the result into theIndex
    if theIndex is -1 then
        answer "There's no color object on the card there."
        exit to HyperCard
    else if theIndex is 1 then
        answer "The object is all the way at the back."
        exit to HyperCard
    end if
    addColor moveBackward,cd,theIndex
    addColor colorCard
end moveBack
```

*Also see:* `moveForward`  
`moveToBack`

## **moveForward**

```
addColor moveForward, {cd|bg}, index
```

---

```
addColor moveForward, cd, 3
```

```
addColor moveForward, bg, theIndex
```

---

This command moves the card or background color overlay at color position *index* one color layer forward.

The overlay's new position is returned in the HyperTalk function `the result`, and will be one number higher than the value in *index*.

The overlay that holds the immediately higher position before this command is executed moves backward one layer. (In effect, the overlays switch layers.)

*Some index values may change:* Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following button handler moves the overlay under the pointer one layer closer to the front.

```
on moveUp
    addColor getObjectClicked,cd,the mouseLoc
    put item 1 of the result into theIndex
    if theIndex is -1 then
        answer "There's no color object on the card there."
        exit to HyperCard
    end if
    addColor moveForward,cd,theIndex
    addColor colorCard
end moveUp
```

*Also see:* `moveBackward`  
`moveToFront`

## **moveToBack**

```
addColor moveToBack, {cd|bg}, index
```

```
addColor moveToBack, cd, 3
```

```
addColor moveToBack, bg, theIndex
```

This command moves the card or background color overlay at color position *index* all the way to the back.

The overlay's new position is returned in the HyperTalk function `the result`. It is always the value 1.

All overlays that hold lower positions before this command is executed move forward one layer.

*Some index values may change:* Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following handler rotates five overlapping button color overlays so that each one in turn momentarily occupies the top position. Hold down the mouse button to exit the handler.

```
on rotateBottomButton
  repeat
    repeat with top = 1 to 5
      addColor getButtonIndex,cd,ID of cd btn top
      addColor moveToBack, cd, the result
      addColor colorCard
      if the mouse is down then exit rotateBottomButton
    end repeat
  end repeat
  beep 3
end rotateBottomButton
```

*Also see:* `moveBackward`  
`moveToFront`

## **moveToFront**

```
addColor moveToFront, {cd|bg}, index
```

---

```
addColor moveToFront, cd, 3
```

```
addColor moveToFront, bg, theIndex
```

---

This command moves the card or background color overlay at color position *index* all the way to the front.

The overlay's new position is returned in the HyperTalk function the `result`.

All overlays that hold higher positions before this command is executed move back one layer.

*Some index values may change:* Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following handler rotates five overlapping button color overlays so that each one in turn momentarily occupies the top position. Hold down the mouse button to exit the handler.

```
on rotateTopButton
  repeat
    repeat with top = 1 to 5
      addColor getButtonIndex,cd,ID of cd btn top
      addColor moveToFront, cd, the result
      addColor colorCard
      if the mouse is down then exit rotateTopButton
    end repeat
  end repeat
  beep 3
end rotateTopButton
```

*Also see:* `moveForward`  
`moveToBack`

## **remove**

```
addColor remove
```

---

```
addColor remove
```

---

This command frees the memory allocated to color for the current stack. It disposes of all the color information stored in off-screen buffers, and immediately turns off all color in the stack.

To see this command in operation, create a stack with some color on it and enter the command `addColor remove` through the Message box.

**Important** This command is automatically installed in the stack script's `closeStack` handler when you first open the color editor. It prevents color artifacts from appearing in the next stack you open.

The following handler deallocates the stack's color memory buffer when the stack is closed.

```
on closeStack
  AddColor remove
  pass closeStack
end closeStack
```

*Also see:* `install`  
`disable`

## removeButton

```
addColor removeButton, {cd|bg}, /D


---


addColor removeButton, cd, 3
addColor removeButton, bg, theID


---


```

This command removes the color overlay from the card or background button whose ID is */D*. The color disappears from the screen at the next call to `colorCard`, `colorBackground`, or `colorCardLayered` (as appropriate).

The index values of all higher-numbered overlays decrease by one.

**Important** This command removes a button's color overlay. It does *not* remove the button. To remove the button, use HyperTalk's `delete` command.

The following handler removes the color overlay from the card button under the pointer.

```
on dumpButtonColor
  put the mouseLoc into thePoint
  -- What button is it?
  repeat with thisOne = number of card buttons down to 1
    if thePoint is within the rect of cd btn thisOne then
      addColor removeButton, cd, id of cd btn thisOne
      addColor colorCard
      exit dumpButtonColor
    end if
  end repeat
  Answer "That's not a button."
end dumpButtonColor
```

*Also see:* `addButton`

### **removeField**

```
addColor removeField, {cd|bg}, /D
-----
addColor removeField, cd, 3
addColor removeField, bg, theID
-----
```

This command removes the color overlay from the card or background field whose ID is */D*. The color disappears from the screen at the next call to `colorCard`, `colorBackground`, or `colorCardLayered` (as appropriate).

The index values of all higher-numbered overlays decrease by one.

**Important** This command removes a field's color overlay. It does *not* remove the field. To remove the field, use HyperTalk's `delete` command.

The following handler removes the color overlay from the card field under the pointer:

```
on dumpFieldColor
  put the mouseLoc into thePoint
  -- What field is it?
  repeat with thisOne = number of card fields down to 1
    if thePoint is within the rect of cd fld thisOne then
      addColor removefield, cd, id of cd fld thisOne
      addColor colorCard
      exit dumpFieldColor
    end if
  end repeat
  Answer "That's not a field."
end dumpFieldColor
```

*Also see:* `addField`

## **removeObject**

```
addColor removeObject, {cd|bg}, index
```

---

```
addColor removeObject, cd, 3
```

```
addColor removeObject, bg, theIndex
```

---

This command removes the color overlay at color layer *index*. The color disappears from the screen at the next call to `colorCard`, `colorBackground`, or `colorCardLayered` (as appropriate). The index values of all higher-numbered overlays decrease by one.

The following handler removes the color overlay under the pointer whether the overlay is on the card or on the background.

```
on dumpColor
  addColor getObjectClicked, cd, the mouseLoc
  get the result
  addColor removeObject, cd, item 1 of it
  addColor colorCard

addColor getObjectClicked, bg, the mouseLoc
  get the result
  addColor removeObject, bg, item 1 of it
  addColor colorCard
end dumpColor
```

*Also see:* `remove`

## sort

```
addColor sort, {cd|bg}
```

---

```
addColor sort, cd
```

```
addColor sort, bg
```

---

This command, which works only in HyperCard 2.2 or later, makes the color layering on the card or background match HyperCard's layering. It arranges objects so that buttons and fields are in front of color rectangles and PICT images in the card and background layers.

**Important** This command bears no relationship to HyperTalk's `sort` command.

If you've reordered color buttons or fields with calls to `moveForward`, `moveBackward`, `moveToFront`, or `moveToBack`, using `addColor sort` properly associates the colors of buttons and fields with their respective objects in the proper layers. The command has no effect, however, on the layering of buttons and fields accomplished through the Bring Closer or Send Farther commands in HyperCard's Objects menu.

*Some index values may change:* Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The following handler sorts the color database for the current card and background.

```
on sortIt
    addColor sort, cd
    addColor sort, bg
end sortIt
```

# Troubleshooting

This appendix describes some common areas of confusion and what to do about them.

*Tip:* To work properly, many new features of HyperCard require a relatively recent version of system software, certain extensions, and sufficient memory allocated to the application. See “System, Software, and Memory Requirements” on page 8 for more information.

## Color

### **There’s no Color menu in the menu bar.**

- Look in the Home stack for a button labeled “Color Tools are OFF.” If you see it, click it; the Color menu will appear in the menu bar.

If you don’t see “Color Tools are OFF,” the Color Tools are probably not yet installed. See Chapter 4, “Installing and Using Color Tools,” for instructions on preparing HyperCard for color.

### **My monitor is set to display thousands of colors, but I can’t see as many colors as I expected in my stack.**

- You need to change the stack’s `openStack` handler. Replace the line `addColor install` with `addColor install, 16` (for thousands of colors).

### **Color doesn’t show in my stacks or stand-alone applications, even though I have a color monitor.**

- Increase the memory allocated to HyperCard and to your stand-alone applications. Begin with an allocation of 2200K (5120K on a Power Macintosh computer) and increase the allocation by 300K until you can see color. For instructions on how to increase memory allocation, see “Preparing a Color Stand-Alone Application” on page 35.

### **I selected an object in the color editor, but Item Info isn’t available.**

- The object hasn’t had color added to it yet. Click a color in the palette; then try again.

**I can't select an item in the color editor.**

- You may not have selected the proper icon at the top of the color palette. Click the Button, Field, Pict, Rect, or Paint icon on the color palette; then select an item of the corresponding type.

**I turned Color Tools off in the Home stack, but I still see color in my stacks.**

- Turning Color Tools off doesn't remove color from a stack. When you add color to a stack, the color becomes a permanent part of the stack unless you remove the stack's color resources with an application program such as ResEdit.

**I tried to print a color stack on my color printer, but none of the colors show.**

- HyperCard doesn't support printing in color. However, you can purchase software that allows you to print your color stacks.

**When I move a color object, the color doesn't follow it.**

- Choose Redraw Screen from the Color menu. If you're moving the object under script control, lock the screen before moving it; then recolor the object with a dissolve effect to make the transition smoother.

**I added color to a stack and immediately saved it as a stand-alone application. When I launch the application, it doesn't show any color.**

- Resources added to a stack are not installed until the stack is closed. After you color a stack, you need to close and reopen it before you save it as a stand-alone application. Make sure the stand-alone application has at least 2200K of memory assigned (5120K of memory on HyperCard for Power Macintosh).

**Color doesn't appear after I add it to my stacks.**

- You probably have an extra `lock screen` call.

**The black-and-white images I replaced with color images appear jerky.**

- Make sure you've replaced HyperCard's visual effect commands with color transition effects. See Chapter 5, "Color Editor Basics," for more information.

## **Opening Files**

**I can't open certain types of files by dragging them onto the HyperCard application icon in the Finder.**

- Try rebuilding your desktop.
- For PICT files, make sure Automatic File Translation is turned on in the Mac OS Easy Open control panel.

**I can't make certain types of files open in my stack.**

- For MPEG files, make sure the QuickTime MPEG extension is installed.
- For QuickTime VR files, make sure QuickTime VR 2.0 or later is installed.
- For QuickTime movies, make sure QuickTime 3.0 or later is installed.

**When I create a new file with the open file command, the icons change for the files opened.**

- When you use this command with QuickTime 3.0 or later and you create a new file with any standard suffix used by QuickTime (such as “.gif” or “.jpeg”), the file will be changed to take the corresponding type and creator, and its icon will reflect this. For instance, files with the suffix “.gif” will show a GIF icon.

To prevent file icons from being changed, open the QuickTime Settings control panel and deselect “Enable QuickTime Exchange,” then restart your computer.

## Images in Stacks

**I chose a PICT image to add to my stack using the Import command, but it doesn't appear.**

- Increase the memory allocated to HyperCard to 4000K or more.

**I get the following message when viewing a stack: “Cannot display this image. There may not be enough memory or the image could not be found.”**

- You may see this message if you place pictures on cards and then delete the picture resource from the stack. Delete each instance of the picture in the stack.

## Button Tasks

**There is an Effects button instead of a Tasks button in the Button Info dialog box.**

- Button tasks are only available on System 7 or later. Also, the Component Manager software is required. Component Manager is controlled by QuickTime and AppleScript. Make sure both the QuickTime and AppleScript extensions are installed and active.

**Tasks are missing from the list in the Button Tasks dialog box.**

- If you see only two tasks—Go to Destination and Visual Effect—then the Tasks file is not installed properly or has been moved. Reinstall the Tasks file into the HyperCard application folder.
- If you see more than two tasks, then the requirements for the tasks that are missing have not been met. (Tasks with additional requirements are Movie, Speak Text, and Link to URL.) See “System, Software, and Memory Requirements” on page 8 for more information.

## Visual Effects

### **My HyperCard visual effects have stopped working.**

- The color editor adds a `closeCard` HyperTalk handler with a `lock screen` command to the stack. This command prevents HyperCard's standard visual effects from working. While HyperCard's visual effects and the Color Tools' transition effects both work if you remove the `lock screen` command, you probably won't like the results. HyperCard's visual effects work only on HyperCard's black-and-white images, with all color turned off as they operate; if you follow a visual effect with any color effect, the image stutters as color is turned on.

## Scripts for Play, Rewind, and Pause

### **My scripts for Play, Rewind, and Pause don't work.**

- This can happen if you have installed an earlier version of the movie command set into your current stack or your Home stack. Remove the older versions of the movie command set from your stacks; it is now included in the HyperCard application.

## Other

### **The HyperCard stack won't open to its full size, and I can't use color or print.**

- Try increasing the memory allocation for HyperCard. If you still experience problems and you have System 7.5 or later, try turning off Automatic File Translation in the Mac OS Easy Open control panel.

### **I can't save my stack as an application.**

- This feature requires the Component Manager software. Component Manager is controlled by QuickTime and AppleScript. Make sure both the QuickTime and AppleScript extensions are installed and active.

### **HyperCard seems to be running slowly.**

- If you've added color to your Home stack, remove it. If not scripted correctly, a color Home stack will cause HyperCard to run slowly.

### **I can't shut down the computer.**

- You must close the color editor or quit HyperCard before using the Shut Down command.

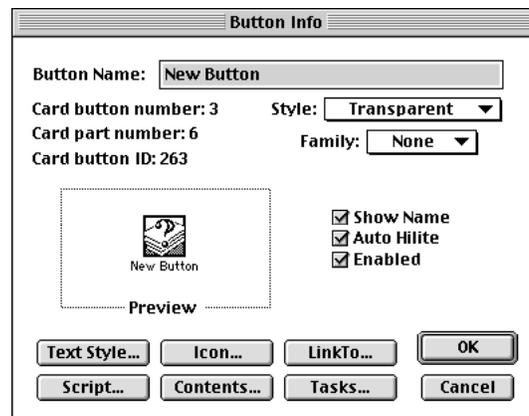
## Updates

This appendix corrects information that has changed since the rest of the documentation was printed and provides information about features that have been improved since the last version of HyperCard was released.

### HyperCard

#### Button Info dialog box

- On computers using System 7 or later, the Button Info dialog box has been changed to reflect the addition of the Tasks button, described in Chapter 1.



- The *HyperCard Reference Manual* states on page 3-22 that this dialog box displays an Effect button. This is true only if you hold down the Option key while you open the Button Info box by double-clicking a button with the button tool. Otherwise, you can add a visual effect by clicking the Tasks button in the Button Info dialog box, and clicking the Visual Effect task.

### **Keyboard shortcuts**

Pressing the tilde key (~) no longer moves you back one card in a stack; instead, pressing ⌘-tilde takes you back.

### **Preview Button in Stack Info dialog box**

The Stack Info dialog box, which appears when you choose Stack Info from the Objects menu, contains a Preview button. Use Preview to change the preview image that appears when Open Stack is chosen from the File menu.

To change the preview image, click Preview and select a card, the Clipboard, or None to show no preview image, from the Source pop-up menu.

## **HyperTalk**

### **answer folder prompt**

The new parameter `folder` displays a dialog box from which you select a folder. The path to the folder, including the trailing colon, is returned in the local variable `it`.

### **blindTyping**

The *HyperCard Script Language Guide* says that the user level must be set to 5 in order to set the `blindTyping` property. In fact, this property can be set at any level.

On the Preferences card in the Home stack, however, you can set this property only if the user level is set to 5.

### **the diskSpace [of disk diskName]**

Formerly, you could get the space available only for the current disk. Now you can specify any mounted hard disk or floppy disk.

### **fade**

The card “Tip:List of synonyms” in the Help Extras stack lists `fade` as a synonym for the visual effect `dissolve`. This is inaccurate.

A similar error appears on page 219 of the *HyperCard Script Language Guide* in the sample script for `go`.

### **find international**

This form of the `find` command is no longer case-sensitive.

### **mouseDoubleClick**

Checkboxes and buttons no longer receive `mouseDoubleClick` messages. Instead they receive `mouseDown`, `mouseStillDown`, and `mouseUp` messages just as they did in versions earlier than HyperCard 2.2.

**open file**

You are no longer limited to a maximum of three open files at one time. See Chapter 3, “Using the New Syntax and Scripting Features,” for additional information on the `open file` command.

**quit system message**

Page 137 of the *HyperCard Script Language Guide* says that `quit` is sent to the current card immediately following the `closeStack` message when you choose Quit HyperCard from the File menu.

This is true only when a single stack is open. When more than one stack is open, `quit` is sent to the current card in the last open stack immediately following that stack's `closeStack` message.

**the stackSpace**

The description of the `stackSpace` function on page 346 of the *HyperCard Script Language Guide* is inaccurate. Instead, use the information in the HyperTalk Reference stack.

**there is a file**

Information regarding `there is a file` on page 116 of the *HyperCard Script Language Guide* is incomplete. When `there is a file` is used, HyperCard searches in the folder containing the HyperCard application.

**the [long] version of [scriptingLanguage] componentName**

The information on `version` in the HyperCard AppleScript Reference stack is incomplete.

If you use the option `long`, this function returns an 8-character hexadecimal string that represents the version of the software and the version of its application programming interface (API). For AppleScript 1.1, the hex string is 01100110.

If you do not use the option `long`, this function returns the version of the component as a decimal string. For AppleScript 1.1, the decimal string is 1.10.

You can compare the returned value to an arbitrary number. For example:

```
if the version of "AppleScript" >= 1.1 then...
```

It is no longer necessary to use `scriptingLanguage` when referring to AppleScript.

**HyperTalk Beginner's Guide**

The *HyperTalk Beginner's Guide*, referred to on page xix of the *HyperCard Script Language Guide*, is no longer available.



## Quick Color Tutorial

This appendix teaches the essentials of using color in a stack. After you've completed this tutorial, you'll know how to

- open the color editor
- use several icons on the color palette
- add color to buttons and fields
- create a color backdrop
- move colors to their proper layers

*What you should already know:* Before you take this tutorial, you should have read the Preface to this book, and you should understand the elements of HyperCard described in Chapter 1 of the *HyperCard Reference Manual*.

### **Making Sure Color Tools Are Installed**

Before you begin, make sure that Color Tools are already installed in your Home stack, and that they are active. Follow these steps:

- 1** Look in the menu bar for a menu named Colors.  
If you see this menu, Color Tools are installed. Go to the next section, "Preparing to Add Color" on page 136. If you don't see this menu, proceed to step 2.
- 2** Look on the first card of the Home stack for an icon labeled "Color Tools are OFF" and click it.



Color Tools are OFF

The icon's label changes to "Color Tools are ON," and you see the Colors menu in the menu bar. Go to the next section, "Preparing to Add Color."



### Color Tools are ON

If you don't see the Color Tools icon, proceed to step 3.

- 3 Install the Color Tools by following the instructions in Chapter 4, "Installing and Using Color Tools."

When the tools are installed, continue with this tutorial.

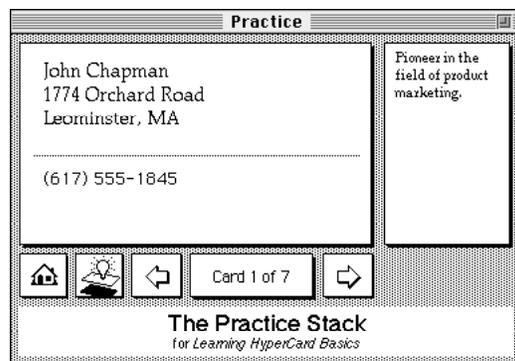
### Preparing to Add Color

In this tutorial you'll be using a copy of the Practice stack. You'll work with the copy so that the original will be preserved in case you want to start over.

To create and open a copy of the stack, follow these steps:

- 1 Click the Practice icon on the first card of the Home stack.

The Practice stack opens.



- 2 Choose "Save a Copy" from the File menu.

A dialog box appears.

- 3 Type `Color Practice` into the name field and click Save.

- 4 Open the Color Practice stack.

Choose Open Stack from the File menu and locate the Color Practice stack. Then click Open.

The Practice stack closes, and the Color Practice stack opens.

## Opening the Color Editor

When you work in color, you leave the standard HyperCard environment and enter the color editor.

- **To open the color editor, choose Open Coloring Tools from the Color menu.**

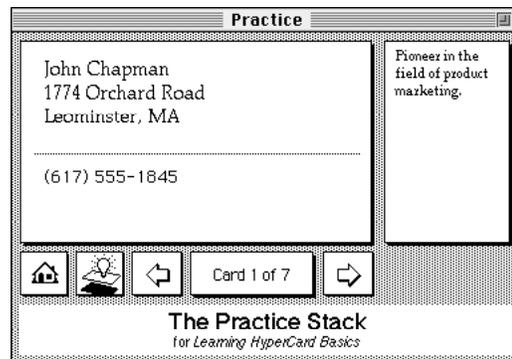
A dialog box appears, asking you to confirm that you want to add color resources to the stack. Click OK.

Any palettes open in HyperCard disappear, and the color editor opens. The color palette appears on the screen.

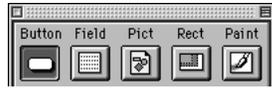


## Adding Color to Buttons

In this exercise, you'll add color to the Previous Card and Next Card buttons.



- 1 Click the Button icon at the upper-left corner of the color palette.



- 2 Click the Previous Card button to select it.
- 3 Click a light color on the color palette to select it.

The button takes on the color you choose.

A dark color may obscure the design of the button. If the color you choose is too dark, click another color.

- 4 Click the Next Card button to select it.
- 5 Click the same color on the palette as you did for the Previous Card button.

The color has a small box around it.



Both buttons now have the same color.

### **Adding a Color Backdrop**

In this exercise you'll add a color backdrop to the background of the stack.

- 1 Choose Background from the Edit menu to open the background.
- 2 Double-click the Rect icon at the top of the color palette.

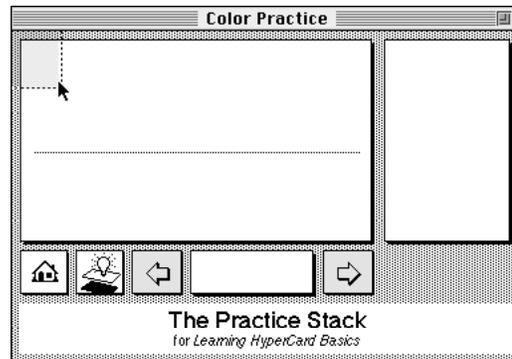
A small square appears on the screen in the current color.

- 3 Click a light color on the color palette.  
Select a color different from the one you used for the buttons, but make sure that it's light.
- 4 Drag the rectangle by its center to the upper-left corner of the stack.  
You can drag the rectangle when the pointer is replaced by a hand.



- 5 Size the rectangle so that it fills the entire background.

To size the rectangle, grab it by its lower-right corner and drag diagonally to the lower-right corner of the stack.



The color of the backdrop you just created obscures the color of the buttons.

- 6 Choose Send To Back from the Items menu.

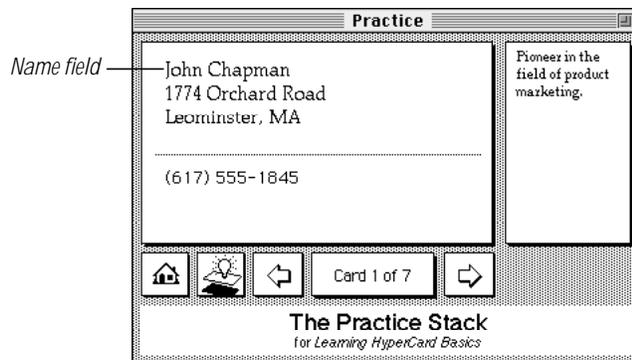
The backdrop moves behind the buttons, and the button color shows over the backdrop.

- 7 Choose Background from the Edit menu to close the background.

### Adding Color to Fields

Now you'll add color to the fields.

- 1 Click the Field icon at the top of the color palette.
- 2 Click the field that has the name in it to select it.



- 3 Click a different light color on the color palette.

It's especially important to choose a light color for a field so that the color doesn't obscure the field's text.

If you don't like the color that appears on the field, click another color.

- 4 If you want, add color to the other fields on the card.

### **Closing the Color Editor**

Now you'll close the color editor and check how the color looks as you change cards.

- 1 Click the close box on the color palette to close the color editor.

You return to the standard HyperCard environment.

- 2 Click the Next Card button.

The color remains as you change cards. All the buttons and fields you added color to are in the background, so the color remains visible.

If you decide you want to change any of the colors you've used, reenter the color editor, select the proper icon on the palette, click the element whose color you want to change, and then click a new color.

You can also return to the color editor and add color to other buttons or fields in the stack, or add smaller color rectangles to highlight different parts of the stack.

### **Where to Go From Here**

To learn more about the color editor, see Chapter 5, "Color Editor Basics."

To learn how to create color pictures, see Chapter 6, "Working With Pictures and Color Paint Tools."

Or you can go back into the color editor and experiment on your own with the Color Practice stack.

# Index

## A

- AC\_RemoteInstall scripting command 76
- addButton scripting command 77
- addColor XCMD 74–126
- addField scripting command 78
- addPictFile scripting command 80
- addPict scripting command 79
- addRect scripting command 81
- aliases for PICT files 54
- Anti-Alias command 65
- Apple Events, new features for 30
- AppleScript
  - Picture XCMD and 29
  - system requirements for 9
- applications
  - reducing memory allocated to 35
  - saving stacks as stand-alone applications 35–36
- Automatic File Translation option, HyperCard problems and 130

## B

- background color, adding 58, 65, 138–139
- Background command 41, 138
- Background pop-up menu 58
- Background Transitions command 43
- Back option 15
- Bevel pop-up menu 48
- bevels, adding to objects 48, 49
- borderless shapes
  - drawing 63
  - finding 46

- Bring Closer command 42
- Bring To Front command 42
- browser. *See* Web browser
- brush tool 61
- built-in button tasks, assigning 13–18
- Button's Current Destination option 15
- Button icon on color palette 44, 138
- Button Info dialog box 14, 131
- buttons
  - adding color to 36–38, 46, 137–138
  - assigning tasks to 13–18
  - assigning visual effects to 15
  - changing position and size of 48
  - creating 13
  - finding 46
  - navigating with 15
  - viewing scripts for 14
- button tasks 129
  - assigning 13–18
  - requirements for 8
- Button tasks window 13–14
- Button tool 13

## C

- cards
  - adding color to 45–46
  - changing 37
  - navigating between 15, 37
  - placing PICT images on 53–54
- Card Transitions command 43
- Centered on Screen movie option 17
- changeObjectBevelscriptingcommand 82

- changeObjectBounds scripting
  - command 83
- changeObjectColorscriptingcommand 84
- changeObjectTransparency scripting
  - command 85
- circles, drawing 64
- Clipboard
  - copying to 49
  - displaying contents of 26
- clipboard container 26
- Close After Playing option 17
- closeCard handler 38, 43
- closed shape tools 63–64
- closeStack handler 38
- closing
  - color editor 40, 130, 140
  - Info box 36
  - movie files 17
  - PICT editor 71
- color
  - adding to background 58, 65, 138–139
  - adding to buttons 36–38, 46, 137–138
  - adding to cards 45–46
  - adding to fields 36, 46, 139–140
  - adding to foreground 58, 65
  - adjusting in Info box 47
  - changing to opposite end of color
    - spectrum 65
  - coloring existing stacks 38
  - copying from one object to another 49
  - disk space and 36–37
  - editing 69
  - Home stack and 37
  - memory and 37, 127
  - pasting 49
  - performance and 36–37, 130
  - preparing stacks for 35
  - problems with 37, 127–128
  - removing from objects 49
  - visual effects and 15, 43, 130
- colorBackground scripting command 86
- colorButton scripting command 87
- colorCardLayered scripting
  - command 90–91
- colorCard scripting command 38, 88
- color commands, using in scripts 37–38
- color depth, changing 57
- color editor
  - closing 40, 130, 140
  - opening 35, 39–40, 137
  - problems selecting items in 127–128
  - Shut Down command and 130
  - tutorial 135–140
  - using the color palette 44
  - using the menus 40–44
  - working with 39–49
- colorField scripting command 92
- color layers, description of 40
- Color menu, commands in 44
- color overlays
  - description of 40
  - editing 41, 49
- color paint tools palette
  - brush tool 61
  - closed shape tools 63
  - color pick-up tool 62
  - eraser tool 62
  - icons on 58, 68
  - lasso tool 60
  - line tool 63
  - marquee tool 59
  - opening 57
  - paint bucket tool 61
  - pencil tool 60
  - shortcuts in 68
  - spray can tool 61
  - system requirements for 9
  - text tool 62
  - using 58–64
- color palette
  - coloring button or field with 46
  - creating a PICT with 57
  - opening 44
  - picking a color with 47
  - uses for 44
- color picker, opening 45, 46, 47
- color pick-up tool 62
- colorPictFile scripting command 95–96
- colorPict scripting command 93–94
- colorRect scripting command 97–98

- color stand-alone application, saving stacks
  - as 35–36
- Color Tools
  - changing color depth 57
  - getting started with 36–38
  - installing 33–34, 135–136
  - memory requirements for 33
  - system requirements for 9
  - turning on and off 34–35, 128, 135–136
- Color Tools button 33
- Color Tools stack icon 33
- color tutorial 135–140
- commands
  - See also individual commands*
  - Apple Events 30
  - in Color menu 44
  - in Edit menu 41
  - in Effects menu 43
  - in File menu 40
  - HyperTalk 23–29, 73–126
  - in Items menu 42
  - using color commands in scripts 37–38
- Compact Color Database command 41
- compact scripting command 99
- Compact Stack command 35
- Component Manager 8, 9
- coordinates, setting 48
- Copy command 49
- copying
  - colors 49
  - stacks 35, 136
- corrected and updated documentation
  - notes 131–133
- Create New Picture command 42, 57
- cropping PICT images 56
- Current Card option 15
- Current Stack option 15
- curved shapes, drawing 64
- Cut command 41, 49

## D

- depth scripting command 100
- disableObject scripting command 102
- disable scripting command 101
- disk space, coloring objects and 36–37

- dithering 56
- documentation, updated and corrected
  - information 131–133
- documents. *See* files
- Draggable Window option 17
- Draw Centered command 63, 67
- Draw Filled command 63, 67
- drawing
  - centered on a given point 63, 67
  - closed shapes 63–64
  - free-form lines 60
  - multiple images 64, 67
- Draw Multiple command 64, 67

## E

- edges
  - outlining 65
  - smoothing 65
- Edit Colors command 69
- Edit Gradient command 70
- editing
  - color overlays 41, 49
  - colors in the PICT editor 69
  - patterns 59
  - PICT images 58–64
  - scripts 31
- Edit menu 41, 49
- Edit Picture command 42
- Effects menu 43, 65–67
- enableObject scripting command 104
- enable scripting command 103
- eraser tool 62
- erasing images 62
- Export Graphics command 68
- exporting PICT images 68

## F

- Field icon on color palette 44
- fields
  - adding color to 36, 46, 139–140
  - changing position and size of 48
  - finding 46
- File menu, commands in 40
- files

*See also* image files; movie files; PICT files  
opening 16  
problems opening 128–129  
Read Me file 9  
Fill command 65  
filling  
    solid areas of image 61  
    with current pattern and colors 65  
Finder, opening movie files in 20  
First Card option 15  
first movie expression 27  
Flip command 67  
Floating Window option 17  
foreground color, adding 58, 65  
Foreground pop-up menu 58

## G

getBitsCall scripting command 105  
getButtonIndex scripting  
    command 106–107  
getFieldIndex scripting command 108–109  
Get Info command 36, 55  
getObjectBevel scripting command 110  
getObjectBounds scripting command 111  
getObjectClicked scripting  
    command 112–113  
getObjectColor scripting command 114  
getObjectType scripting command 115  
getPictName scripting command 116  
GIF files 19  
GIF files, opening 16  
Go to Destination button task 15  
gradient, creating 70–71

## H

height of images, changing 55  
Home option 15  
Home stack  
    coloring and 37  
    installing Color Tools into 33–34, 135–136  
    navigating to cards in 15  
HyperCard  
    installing 9  
    learning about 10–11, 135–140

memory requirements for 8, 33  
system requirements for 8  
updated and corrected information 11,  
    131–133

HyperCard Installer icon 9  
*HyperCard Reference Manual* 10, 11  
*HyperCard Script Language Guide* 10  
HyperCard Tour 10

HyperTalk

*See also* scripting functions; scripts  
button tasks and 13  
color commands in scripts 37–38  
new scripting features 23–31  
QuickTime syntax 20  
scripting syntax added by Color Tools 73–126  
updated and corrected information 11,  
    132–133

## I

icons

Button icon 44, 138  
Color Tools stack icon 33  
Field icon 44  
HyperCard Installer icon 9  
on color paint tools palette 58, 68  
on color palette 44  
Paint icon 44  
Pict icon 44, 52, 53, 55  
Rect icon 44

image files

*See also* files; PICT files  
opening with Movie button task 16  
Picture XCMD and 29

images. *See* PICT images

Import Graphics command 68

importing

PICT resources 52–53, 68, 129  
sounds 17

Info box

changing object's position 48  
changing object's RGB numbers 47  
changing object's size 48  
closing 36  
opening 47  
scaling PICT images 55

- using color object's Info box 47–48
- Inside Macintosh* programming guide 31
- installing
  - Color Tools 33–34, 135–136
  - HyperCard 9
- `install` scripting command 38, 117
- Internet Config
  - Link to URL button task and 8, 16
  - `open URL` scripting command and 27
- Invert command 65
- invisible objects, finding 46
- Item Info command 42, 47, 127
- Items menu, commands in 42

**J**

- jagged edges, smoothing 65
- JPEG files 19
- JPEG files, opening 16

**K**

- keyboard controls, in QuickTime VR movies 20
- keyboard shortcuts 132

**L**

- lasso tool 60
- Last Card option 15
- `last movie` expression 27
- layers, description of 40
- line tool 63
- Link to URL button task 8, 16
- `lock screen` scripting command 38, 43

**M**

- Mac OS Easy Open, HyperCard problems and 130
- magnification setting 69
- marquee tool 56, 59
- memory
  - changing for stand-alone applications 35–36
  - optimizing for color stacks 37, 127
  - problems opening stacks and 130
- memory requirements
  - for Color Tools 33

- for HyperCard 8, 33
- for stand-alone applications 36, 127

monitor

- color depth of 9, 36, 57
- color problems and 127

`moveBackward` scripting command 118

`moveForward` scripting command 119

`moveToBack` scripting command 120

`moveToFront` scripting command 121

Movie button task

- opening movie files with 16–17, 19
- system requirements for 8

movie files

- See QuickTime movies; QuickTime VR movies

MPEG files, opening 16, 129

multiple shapes, drawing 64

**N**

- negative indices, support for 30
- New Button command 13
- Next Card option 15
- No Destination option 15

**O**

objects

- adding color to 36–38, 45–46, 137–140
- changing position and size of 48
- copying colors between 49
- finding 46
- removing color from 49

Objects menu 13, 14

ObjectSupportLib file 8, 9

Opaque command 67

`openCard` handler 38

Open Coloring Tools command 35, 44, 137

opening

- applications 15
- Button tasks window 13–14
- color editor 35, 39–40, 137
- color paint tools palette 57
- color picker 45, 46, 47
- files 16, 27, 128–129
- image files 16
- Info box 47

- movie files 16–17, 19–20
- PICT editor 56, 57
- QuickTime movies 16, 129
- URLs 16, 27
- Open Stack command 132
- openStack handler 38
- open URL command 27
- Options menu 59, 61, 63, 67
- outlining edges 65
- oval shapes, drawing 64
- overlays. *See* color

## P

- paint bucket tool 61
- paint effects, setting 64–68
- Paint icon on color palette 44
- palette. *See* color paint tools palette; color palette; system palette
- pass *messageName* call 38
- Paste command 49
- patterns, choosing and editing 59
- Patterns pop-up menu 58, 59
- pencil tool 60
- performance
  - color and 36–37, 130
  - color commands in scripts and 37
  - PICT files and 36
  - problems with 37, 130
  - scripts and 37
- PICT editor
  - closing 71
  - creating a gradient 70–71
  - editing colors 69
  - importing and exporting 68
  - opening 56, 57
  - saving images 69
  - setting paint effects 64–68
  - zooming 69
- PICT files
  - See also* files; image files
  - aliases for 54
  - disk space and 52
  - displaying 54
  - opening 19, 20, 29, 128
  - performance and 36

- Picture XCMD and 29
- troubleshooting 128
- versus PICT resources 51–52
- Pict icon on color palette 44, 52, 53, 55
- PICT images
  - adding to stacks as resource 52–54
  - applying effects to 64–68
  - creating 57
  - cropping 56
  - disk space and 36, 51–52
  - displaying as files 52
  - dithering 56
  - editing 58–64
  - erasing 62
  - exporting 68
  - importing 52–53, 68, 129
  - performance and 36
  - placing on cards 53–54
  - saving 69
  - scaling 55, 66
  - tinting 65
  - troubleshooting 129
  - using the color paint tools palette 58–68
  - zooming 69
- PICT resources
  - disk space and 51
  - displaying in stacks 52–54
  - versus PICT files 51
- picture property 30
- pictures, adding color to 36
- Picture XCMD, new features for 29
- Place Picture command 42, 44, 52, 53, 55
- Place Rectangle command 42, 44
- playing QuickTime movies 19–20
- play scripting command 26
- polygons with irregular sides, drawing 64
- position of objects, changing 48
- Preview button 132
- Previous Card option 15
- problems. *See* troubleshooting

## Q

- QuickDraw picture, returning bitmap as 30
- QuickTime movies
  - closing 17

- integrating into stacks 19–21
- opening in Finder 20
- opening with Movie button task 16–17, 19
- options for displaying 17
- Picture XCMD and 29
- playing 19, 20
- references to 27
- system requirements for 8
- troubleshooting 128, 129, 130
- QuickTime MPEG extension 19, 129
- QuickTime Tools stack 20–21
- QuickTime VR movies
  - Close After Playing option 17
  - closing 17
  - controlling with keyboard 20
  - Draggable Window option and 17
  - integrating into stacks 19–21
  - opening in Finder 20
  - opening with Movie button task 16–17, 19
  - options for displaying 17
  - playing 19–20
  - system requirements for 8
  - troubleshooting 128, 129
  - using one movie to control another 21
- Quit HyperCard command 40

## R

- Read Me file 9
- recording sound 17
- rectangles
  - adding color to 36
  - creating 45
  - drawing 64
- Rect icon on color palette 44
- Redraw Screen command 44, 128
- removeButton scripting command 123
- removeField scripting command 124
- removeObject scripting command 125
- remove scripting command 38, 122
- restoring scaled image to original size and proportions 56
- Revert command 69
- RGB number, modifying in Info box 47
- Rotate command 65

## S

- “Save a Copy” command 35, 136
- Save command 69
- saving PICT images 69
- Scale command 66
- scaling PICT images 55, 66
- script editor window, enhancements to 31
- scripting syntax 23–31, 73–126
  - AC\_RemoteInstall 76
  - addButton 77
  - addField 78
  - addPict 79
  - addPictFile 80
  - addRect 81
  - changeObjectBevel 82
  - changeObjectBounds 83
  - changeObjectColor 84
  - changeObjectTransparency 85
  - colorBackground 86
  - colorButton 87
  - colorCard 38, 88
  - colorCardLayered 90–91
  - colorField 92
  - colorPict 93–94
  - colorPictFile 95–96
  - colorRect 97–98
  - compact 99
  - depth 100
  - disable 101
  - disableObject 102
  - enable 103
  - enableObject 104
  - getBitsCall 105
  - getButtonIndex 106–108
  - getFieldIndex 108
  - getObjectBevel 110
  - getObjectBounds 111
  - getObjectClicked 112–114
  - getObjectColor 114
  - getObjectType 115
  - getPictName 116
  - install 38, 117
  - lock screen 38
  - moveBackward 118
  - moveForward 119

- moveToBack 120
- moveToFront 121
- new scripting features 23–31
- remove 38
- removeButton 123
- removeField 124
- removeObject 125
- sort 126
- syntax terms and conventions 73–74
- scripts
  - See also* HyperTalk; scripting syntax
  - adding with Button tasks window 13–14
  - appearance of script editor menus 31
  - editing 31
  - navigating through 31
  - performance and 37
  - script editing window 31
  - toolkit for 20
  - troubleshooting 128, 129, 130
  - using color commands in 37–38
  - viewing scripts for button tasks 14
- Send Farther command 42
- Send To Back command 42
- shading, adding to objects 48
- shortcuts
  - keyboard shortcuts 132
  - paint effects shortcuts 68
- Show Movie Controller option 17
- shutting down the computer 130
- size of objects, changing 48
- software requirements. *See* system requirements
- Sort Color Database command 41
- sort scripting command 126
- sound
  - HyperCard support for 25
  - importing 17
  - recording 17
  - sound feature (HyperTalk) 25–26
  - system requirements for 25
- Sound button task 8, 17
- speak scripting command 24
- Speak Text button task 8, 18
- speech
  - scripting 24–25
  - Speak Text button task 18
  - system requirements for 8
- Speech Manager software 8
- speed settings for transition effects 43
- spray can tool 61
- squares, drawing 64
- Stack Info dialog box 132
- stacks
  - adding color to objects in 36–37, 137–140
  - adding PICT images to 52–54
  - coloring existing stacks 38
  - compacting 35
  - copying 35, 136
  - installing Color Tools in Home stack 33–34, 135–136
  - integrating QuickTime movies into 19–21
  - memory and 37, 127, 130
  - navigating between cards in 15, 37
  - preparing for color 35
  - problems opening images in 129
  - problems with colored Home stacks 37
  - QuickTime Tools stack 20–21
  - reducing size of 35
  - saving as applications 9, 35, 130
  - transition effects in 43
  - using color efficiently in 37–38
- Stack Transitions command 43
- stand-alone applications
  - changing memory allocation of 35–36
  - memory requirements for 36, 127
  - saving color stacks as 35–36
  - saving stacks as 130
  - troubleshooting 127, 128, 130
- stop sound scripting command 26
- stop speech scripting command 25
- straight lines, drawing 63
- system palette 51, 57
- system requirements
  - for all features 8
  - for AppleScript 9
  - for button tasks and HyperTalk commands 8
  - for Color Tools 9
  - for QuickTime features 8
  - for sound 8, 25
  - for speech 8
  - to save stacks as applications 9

## T

- tasks, assigning to buttons 13–18
- text, adding to images 62
- text tool 62
- text-to-speech feature (HyperTalk) 24–25
- the `clipboard` container 26
- the `number of movies` expression 27
- the `soundChannel` property 25–26
- the `speech` function 24
- the `voices` function 25
- three-dimensional effects 48
- Tint command 65
- tinting images 65
- “Top Left Corner at” option 17
- Trace Edges command 65
- transition effects, color stacks 43
- Transparent command 67
- troubleshooting 127–130
  - button tasks problems 129
  - colored Home stacks and 37
  - Color menu doesn’t appear 127
  - color problems 37, 127–128
  - distorted speech 18
  - images in stacks 129
  - Item Info not available 127
  - monitor doesn’t display enough colors 127
  - opening files 128–129
  - performance problems 37, 130
  - saving a stack as an application 130
  - scripts for Play, Rewind, and Pause 130
  - selecting item in color editor 128
  - shutting down computer 130
  - stand-alone applications 127, 128
  - visual effects 130
- turning Color Tools on and off 34–35, 128, 135–136
- tutorial, color 135–140

## U

- Undo command 41, 49
- updated and corrected documentation notes 11, 131–133
- URLs
  - opening 16, 27
  - system requirements for 8

## V

- `version` function 28
- virtual memory, Color Tools and 9
- Visual Effect button task 15
- visual effects
  - assigning to buttons 15
  - color and 15, 43, 130
  - transitions and 43
  - troubleshooting 130
- voices
  - `speak` scripting command 24
  - Speak Text button task 18

## W

- Web browser
  - launching with Link to URL button task 16
  - launching with `open URL` command 27
- width of images, changing 55

## Z

- Zoom box on color paint tools palette 68
- Zoom In command 69
- zooming images 69
- Zoom Out command 69