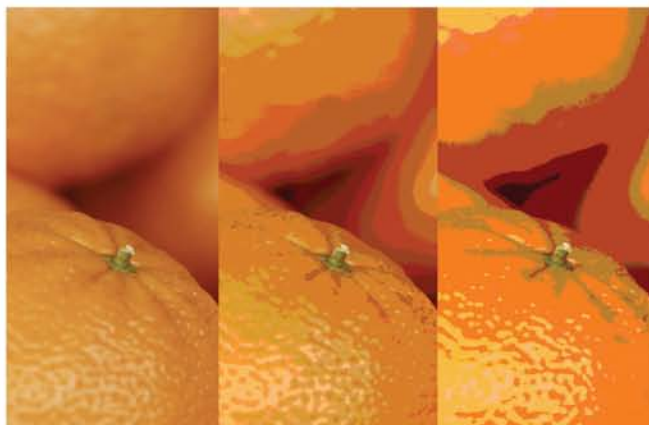
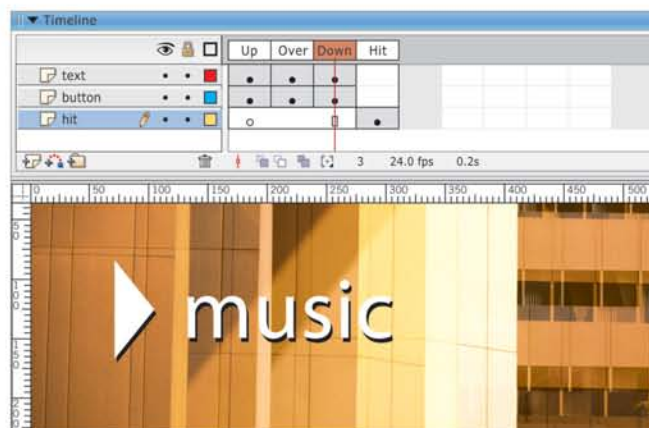


# Flash 8

## *Projects for Learning Animation and Interactivity*



*Flash 8 is a powerful but complex program for creating web-based animations and interactive features. This book demystifies Flash, letting you tap its power quickly with real, hands-on projects.*

## Flash 8: Projects for Learning Animation and Interactivity

If you're new to Flash—and especially if you're coming from a graphics background—then *Flash 8: Projects for Learning Animation and Interactivity* is for you. This book teaches you Flash with step-by-step projects that help you create actual content as you go.

Each chapter provides graphically appealing, hands-on examples that teach by illustration, not with text-heavy description. And each project in the book grounds you in design concepts and goals before moving on to implementation and scripting.

Coauthored by Rich Shupe and Robert Hoekman, Jr., educational developers with years of experience creating compelling content, interfaces, and applications, *Flash 8: Projects for Learning Animation and Interactivity* is not only content-driven, it's also inspiration-driven.

**Understand Flash concepts quickly**

**Accomplish tangible goals in creative ways**

**Create everything from simple animations to interactive elements and custom web site interfaces**

**Enhance projects with graphics, sound, and video**

**Master animation techniques such as tweening, morphing, and dynamic masking**

**Learn ActionScript to take control of your content**

**Develop Flash content for e-Learning, CD-ROM, and mobile devices**



**Included CD-ROM contains comprehensive step-by-step source files and trial software**

**O'REILLY®**

digitalmedia.oreilly.com  
www.oreilly.com

US \$34.99

CAN \$48.99

ISBN: 978-0-596-10223-4



5 3 4 9 9

**Safari®**  
BOOKS ONLINE  
ENABLED

**Includes  
FREE 45-Day  
Online Edition**

*Flash 8: The Missing Manual* is a great companion to this book. Check it out at [www.oreilly.com](http://www.oreilly.com) or your favorite local or online bookstore.



**Praise for the authors of *Flash 8: Projects for Learning  
Animation and Interactivity***

“Rich Shupe can be admired for his encyclopedic knowledge of all things Flash, but it is the combination of creative ingenuity and teacher’s intuition that make him a writer of rarefied skill. When Rich talks about Flash he manages to strip the mystery out and replace it with imagination. He is, at once, artisan and guru.”

—Lori Piquet, *Editor-in-Chief*, DevX

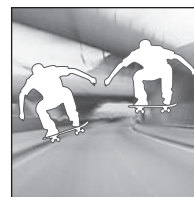
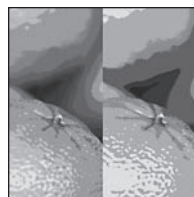
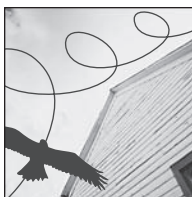
“Hoekman’s approach is clear and unpretentious—attributes not always associated with software how-to books. He leaves the reader who actually ventures into the lessons with the feeling of ‘Hey! I can do this!’ It’s a nice bonus that he doesn’t show off his prowess in the jargon department.”

—Michael W. Dominowski, *Staten Island Sunday Advance*



# Flash 8

## *Projects for Learning Animation and Interactivity*



Rich Shupe and Robert Hoekman, Jr.

O'REILLY®

BEIJING • CAMBRIDGE • FARNHAM • KÖLN • PARIS • SEBASTOPOL • TAIPEI • TOKYO

# Flash 8: Projects for Learning Animation and Interactivity

by Rich Shupe and Robert Hoekman, Jr.

Copyright © 2006 O'Reilly Media, Inc. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles ([safari.oreilly.com](http://safari.oreilly.com)). For more information, contact our corporate/institutional sales department: 800-998-9938 or [corporate@oreilly.com](mailto:corporate@oreilly.com).

## Print History:

March 2006: First Edition.

**Editor:** John Neidhart

**Production Editor:** Genevieve  
d'Entremont

**Copyeditor:** Rachel Wheeler

**Proofreader:** Sada Preisch

**Indexer:** Johnna VanHoose Dinse

**Cover Designer:** Linda Palo

**Interior Designer:** David Futato

**Illustrators:** Robert Romano,  
Jessamyn Read, and Lesley Borash

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. The Digital Studio series designations, O'Reilly Digital Studio, *Flash 8: Projects for Learning Animation and Interactivity*, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 0-596-10223-2

ISBN13: 978-0-596-10223-4

[C]

[02/07]

# Contents

<b>Preface .....</b>	<b>ix</b>
<b>1. Getting Started, Right Out of the Box .....</b>	<b>1</b>
Drawing Your First Box	1
Coloring Fills and Strokes	7
Merging and Stacking Shapes	16
Creating Reusable Graphics	22
<b>2. Creating Quickly: Customizing Your Workspace .....</b>	<b>27</b>
Designing Your Own Panel Layout	27
Customizing Movie Properties	32
Aligning Objects on the Stage	33
Behind Every Good Symbol Is a Good Editor	37
Automate Your Workflow	39
<b>3. Your First Animation .....</b>	<b>45</b>
Layers and the Timeline	46
Keyframes and Tweening	50
Preparing Text for Animation	52
Staggering Animation	57
Alpha Effect	58
Motion Effects	58
Your First Script	63
Publishing Your Movie	65
<b>4. Buttons and Interactivity .....</b>	<b>69</b>
Buttons as Symbols	69
Scripting Your Button	73
Components and Behaviors	79
Navigation	83
More Fun with Buttons	90

<b>5. Working with Graphics .....</b>	<b>95</b>
Importing Pixels .....	95
Working with Pixels .....	101
Importing Vectors .....	106
Using Scenes .....	109
Working with the Library .....	110
<b>6. Movie Clips and Interactivity .....</b>	<b>115</b>
Drawing a Cartoon Character .....	115
Controlling the Character with ActionScript .....	128
More Movie Clip Control .....	131
<b>7. More Animation Techniques .....</b>	<b>135</b>
Morphing with Shape Tweens .....	135
Frame-by-Frame Animation .....	139
Using Masks .....	143
Timeline Versus ActionScript Animation .....	155
<b>8. Using Sound .....</b>	<b>161</b>
Importing Sounds .....	161
Controlling External Sounds .....	169
Scripting Your Own Sound Control .....	172
<b>9. Using Video .....</b>	<b>181</b>
Importing Video .....	181
Controlling External Videos .....	184
Scripting Your Own Video Control .....	190
<b>10. Compositing and Bitmap Effects .....</b>	<b>197</b>
Runtime Bitmap Caching .....	197
Bitmap Filter Effects .....	199
Blend Modes .....	210
<b>11. Working with Text .....</b>	<b>217</b>
Text Types .....	217
Using Fonts .....	221
Loading and Styling Text .....	223
FlashType .....	229
<b>12. Loading Assets on the Fly .....</b>	<b>235</b>
Using ActionScript to Modularize Content .....	235
Preloading .....	240
<b>13. e-Learning with Flash .....</b>	<b>249</b>
Creating a Quiz: Getting Started with Templates .....	249
Sending Results with a Form .....	254
Saving and Retrieving Local Data .....	260
<b>14. Flash for CD-ROM and Handhelds .....</b>	<b>271</b>
Flash on CD-ROM .....	271
Flash on the Run .....	277

<b>15. Think Outside the Box</b>	<b>291</b>
Exporting to Video	291
The Drawing API	299
Extending Flash	303
The Rest Is Up to You	311
<b>A. Tips and Resources</b>	<b>313</b>
<b>Index</b>	<b>335</b>



# Preface

Welcome to *Flash 8: Projects for Learning Animation and Interactivity*, a project-based introduction to the newest version of Flash. This book includes everything you need to get started in Flash—a robust animation and interactive development tool—right out of the box, including a CD-ROM with trial versions of the software and all of the example files from the exercises herein.

## Who This Book Is For

This book is primarily geared toward beginning Flash users who want to bring their project ideas to life. This is not an exhaustive look at every feature available in Flash, nor is it an ActionScript dictionary. This book offers a different perspective.

*Flash 8: Projects for Learning Animation and Interactivity* has been designed as a project-based introduction to the Flash world. Easy-to-follow exercises, content-rich sidebars, and plenty of illustrations work together to guide you through the application's major features. You will never find yourself overwhelmed by unnecessarily complex exercises or dry, labored discussions. At the same time, however, you will be inspired to create in new and different ways.

This book was written with a few simple assumptions in mind: that even if you're a beginner, you are relatively intelligent and motivated, you have a general familiarity with typical graphics programs and web browsers, and you have some basic HTML experience. However, you do not need to be well versed in ActionScript to enjoy this book. In fact, quite the contrary is true.

*No programming experience is required.* ActionScript fundamentals are slowly introduced throughout the book to help guide you through each new topic. You can take advantage of the sample files supplied on the enclosed CD-ROM and learn to program at your own pace. You needn't fully understand all the code examples right away, but when you feel comfortable, the

## In this chapter

Who This Book Is For

More Than Just Marketing

What Makes This Book Different?

How to Use This Book

Using Code Examples

Safari® Enabled

Comments and Questions

Acknowledgments

ActionScript foundation presented in this book should prepare even the greenest of beginners to take your Flash skills to the next level.

### **Not Exactly a Beginner?**

While essentially an entry-level work, this book is not just useful for the uninitiated. It is also an excellent choice for readers who have a superficial familiarity with Flash and want to learn more about the new features introduced with Version 8.

This book should also be effective for programmers with no prior Flash experience who need to learn the Flash development environment. If, however, you want a more in-depth look at ActionScript, take a moment now to refer to the resource listing in Chapter 15. You may find a pointer to a book better suited to your needs.

If you're looking for a broader focus, and you want to get up to speed in Flash with minimal effort, this book is for you. It follows the success of tutorial instruction, providing information on a need-to-know basis, without muddying the waters unnecessarily.

## **More Than Just Marketing**

The subtitle of this book is more than just a catch phrase, it emphasizes two important themes that run throughout. First, this book will show you how to use Flash right out of the box. You'll dive right in and learn how to animate, play audio and video, program basic interactivity, and more. Second (and just as important), as you make your way through these pages, you'll learn how to think outside the box. The last two chapters, in particular, show you a few examples of how to create with Flash in nontraditional ways.

In this book, a box will serve as a unifying element. It will be the basis for graphic layouts, buttons, animations—even a cartoon character. It will be there to remind you of what is possible without additional resources, as well as what is possible when you think without boundaries.

Boxes are allegorical of many aspects of design and development. A box can suggest form and structure, but it can also represent an empty container that you can fill with ideas. This book will help you develop some of the skills you need to fill that box.

## **What Makes This Book Different?**

Most entry-level books are structured with a rigid, linear sequence of chapters planned with the hope that one topic will logically lead to the next. Sometimes, however, this approach may not cover a subject when you really need it or, perhaps worse, may thoroughly cover a topic in which you have

no interest. By contrast, this book tries to pair topics with goals, making learning the material a more organic process. Much like peeling an onion, this book will expose new concepts layer by layer, sometimes revisiting a point more than once to make it easier to grasp.

Several techniques are used in this book to help you better understand and retain the information you are given. This approach is what makes this book a more effective learning tool than other books on the shelf. Here's a breakdown of the approach:

#### *Learning by doing*

People learn by repetition, so you should perform all the exercises contained in this book. Techniques presented in one chapter are often used again in later chapters, offering practice while you learn new concepts. Subsequent uses will require less description and tutorial focus, until eventually accomplishing basic tasks becomes second nature. You aren't expected to master skills at the first use, and this approach means that you won't have to absorb too much information at once.

#### *Shortcuts*

Shortcuts and alternative techniques are introduced after you have performed an operation at least once. Becoming more familiar with each operation from multiple perspectives reinforces the material, helping you to transfer your newly acquired knowledge from short-term to long-term memory.

#### *Chunking*

Information in this book is grouped together in small digestible parts. This process is known as *chunking*. To see how this works, try to remember the following nine-letter sequence: *pnggifjpg*. Not too easy. Now try to remember the chunks PNG, GIF, and JPG. Simple, eh? The difference is in the presentation. Chunking makes it easier for you to understand broader ideas, instead of just repeating things by rote.

#### *Need-to-know basis*

As previously mentioned, this book doesn't try to teach you everything about a given topic before you're ready. Instead, it features "progressive disclosure," in which a topic is revisited in more depth as your knowledge and needs grow. For example, don't be concerned if a specific chapter doesn't describe all of the settings in a particular dialog. You may learn about portions of the dialog in one discussion and revisit the options in the remainder of that dialog later. This reduces the chance that you'll feel overwhelmed.

#### *Sidebars*

In addition to presenting new information, sidebars are also used to expand upon topics introduced in the main text. Subjects are usually explained in the text just enough to convey the main ideas, or for you to perform an exercise. If warranted, additional detail may appear in a

sidebar. This allows more experienced readers to move on without interruption, while beginners can learn in stages (but see “A Few Important Words About Sidebars,” later in this Preface, for a caution).

### *Suggesting the next step*

Each chapter ends with a section called “What’s Next?” This conclusion not only prepares you for the upcoming chapter, but also offers simple suggestions for you to expand on what you’ve learned. These unguided goals push you to develop related skills by accomplishing tasks *without* numbered steps or sample files. They are designed to motivate you into taking your skills to the next level and to start you down the road of self-teaching.

## How to Use This Book

If you read this book from cover to cover, you’ll find many tidbits in unlikely places that you’d miss if you skipped around. You’re strongly encouraged to perform all the exercises, even if they don’t appear to relate to tasks that you specifically want to accomplish, in order to gain familiarity with Flash. Most of the exercises build on concepts and operations learned earlier, so unless you’re already familiar with Flash, you should start at the beginning. The exercises are refreshingly brief, so give them a shot and pick up the finer points hidden along the way.

### Getting Started

Nothing is more frustrating than a tutorial book in which you can’t get the examples to work. If you run into trouble, bear the following in mind.

Flash 8 is sold in two versions: Basic and Professional. Think of the Professional version as a full product and the Basic version as an entry-level, or functionally limited, product. (See <http://www.macromedia.com/software/flash/basic/> for a comparison of the two versions.) Many of the basic features are common to both versions, but the two start to diverge when it comes to more in-depth capabilities. To cover the most ground, this book and some of its sample files assume you are using the Professional version, but you can still get a lot from these pages if you’re using Flash 8 Basic. Installers for both versions are included on the enclosed CD-ROM. If you don’t already have Flash 8 installed, it’s a good idea to evaluate the Professional version.

#### NOTE

*If you have lost the CD-ROM and have a high-speed connection, you can download the trial software from Macromedia’s web site (<http://www.macromedia.com>). You can also download the sample files, as well as any updates or corrections, from the book’s support web site at <http://www.flash8projects.com>.*

## A Few Important Words About Sidebars

The use of sidebars in this book differs from their use in many other books, in that they do not merely contain supplemental material that tangentially relates to the topic at hand. Instead, sidebars are used to pull important or lengthy topics out of the body text, where they might otherwise be too weighty or significantly interrupt the flow of the chapter. This is especially true in the step-by-step projects.

Therefore, it is highly recommended that you read every sidebar. ActionScript topics, in particular, are introduced or expanded upon throughout the book in sidebars. This book has been designed to allow you to work through the projects quickly and to learn by doing. Rather than dedicating a full chapter or more to ActionScript theory, we introduce these topics as you go, in smaller pieces that are easier to digest and are presented just when you need them most.

Again, you are highly encouraged to embrace sidebars as content that is as valuable as the main body of the text. Skipping sidebars because you assume they are of lesser importance will almost certainly reduce the effectiveness of this book.

## Conventions Used Herein

The following typographical conventions are used in this book:

### *Keyboard shortcuts*

Windows and Mac OS keyboard shortcuts are often listed for commands, especially the first few times they are mentioned. If only one shortcut is specified, it is the same on both platforms. Typically, the shortcuts only differ because they begin with the Control key on the Windows platform and the Command key on the Macintosh platform. These modifier keys are usually cited as “Ctrl-G (Win) or Cmd-G (Mac).” For brevity, they may be listed as “Ctrl/Cmd-G.”

### *Menu commands*

Menu commands are indicated using the arrow symbol (→). For example, Edit→Copy indicates that you should select the Copy command from the Edit menu. The same convention is used to indicate that you should choose a tab or suboption in a dialog box, such as File→Publish Settings→Flash→ActionScript Version.

### *Italic*

Indicates new terms, ActionScript references (names of functions, methods, parameters, properties, etc.), symbol names, symbol linkage identifiers, frame labels, and URLs, as well as the names of layers, files, directories, and similar items requiring emphasis.

**Constant width**

Indicates code samples, named instances of movie clips, XML tags, HTML tags, the contents of files, or the output from commands.

**Constant width bold**

Shows commands or other text that should be entered literally by the reader. This font is also sometimes used within code examples for emphasis, such as to highlight an important line of code in a larger example.

**NOTE**

*This icon signifies a tip, suggestion, or general note. Many tips are integrated throughout the text, but notes set aside in this manner may be particularly important.*

**WARNING**

*This icon indicates a warning or caution. Ignore it at your own peril.*

## Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Flash 8: Projects for Learning Animation and Interactivity*, by Rich Shupe and Robert Hoekman, Jr. Copyright 2006 O'Reilly Media, Inc., 0-596-10223-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).

## Safari® Enabled



When you see a Safari® Enabled icon on the cover of your favorite technology book, that means it's available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books: it's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

## Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
(800) 998-9938 (in the United States or Canada)  
(707) 829-0515 (international or local)  
(707) 829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/flashprojects>

To comment or ask technical questions about this book, send email to:

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

<http://www.oreilly.com>

## Acknowledgments

Rich Shupe would like to thank the following people for their help, support, and distractions during his work on this book:

- John Neidhart, Steve Weiss, and everyone at O'Reilly Media, as well as Robert Hoekman, Jr., and Bruce Epstein
- Jodi Rotondo, Thomas Yeh, Kawai Sin, Joseph Shoemaker II, and everyone at FMA, including my students
- Bruce Wands, Joe Dellinger, Russet Lederman, Mike Barrons, Jaryd Lowder, Diane Field, and everyone at the School of Visual Arts, including my students
- Lori Piquet and everyone at DevX
- Stewart McBride, Lynda Weinman, Kevin Skogland, Christoph Wiese, Gaylynn Firth, Kim Vandyk, and everyone at FlashForward

- Paul Kent, Rachael Jones, Kate Greene, Heather Meninno, and everyone at Macworld, IDG, and Mactivity
- Steve, Cindy, and Brian Shupe; Abigail Jannsens; Dennis, Elaine, Denise, and Doug Rotondo; and Cheri Strand
- The Jungle (especially Penn Jillette and Teller), The Residents, Tony Fitzpatrick, Tim Jenison, Emily Z. Jillette, Paul Provenza, and all aristocrats everywhere
- Extra special thanks to Jodi, Thomas, Brian, and Jesse Freeman; and welcome Kaito, Enzo, Moxie, and...?

Robert Hoekman, Jr., would like to offer his sincerest gratitude to the following people:

- Bruce Epstein
- O'Reilly Media, Inc. (especially Tim O'Reilly, Rob Romano, Claire Cloutier, Glenn Bisignani, and Norma Emory)
- Robert Eckstein (and Scrappy)
- Liatt Bailey
- Beta readers Sham Bhangal, Paul Catanese, Lisa Coen, Marc Garrett, Mark Jonkman, Andy Rayne, Darron Schall, Drew Schiffman, Dana Stokes, Karen Vagts, and Edoardo Zubler
- Macromedia (in particular Mike Downey, Mike Chambers, Ed Sullivan, and Amy Brooks)
- FMUG.az (Flash and Multimedia Users Group of Arizona): John C. Bland II, Shane Anderson, Bob Wohl, Ron Haberle, Muharem Lubovac, Shaun Jacob, Jeff Garza, and everyone else in the group
- Above all else, my wife, Christine Rose Pearson

# Getting Started, Right Out of the Box

# 1

In this chapter, you'll gain your first experience with Flash's vector drawing tools. Many of Flash's tools are similar to those available in other applications, so you may be able to apply some of what you already know to Flash. However, some of Flash's tools behave in unique ways. For example, Flash builds everything you draw using vectors. So, although using Flash's Brush tool will feel similar to using a brush tool in a bitmap editing application such as Photoshop, what you draw will actually be a vector shape.

Flash's selection tools also behave a bit differently than those in a typical drawing application. You can select a shape's fill or stroke independently, and even separate the two. Also, using the Lasso tool, you can freely draw a selection path across any shapes and work only with the selected portion of those shapes. This combination of pixel- and vector-based drawing approaches makes creating Flash assets a natural and immediate experience.

If you are comfortable with traditional vector drawing applications, don't worry; you will also find a Pen tool for creating Bezier curves complete with vertices and control points in Flash. However, even if you are unfamiliar with these terms, you'll still be able to jump right in and start creating.

Your first project is to create a graphic you might use in a psychedelic poster illustration of guitar legend Jimi Hendrix. You'll begin by drawing a *box*, which is one of the most commonly used shapes in design and a frequently used metaphor in this book. Next, you'll customize your lines and fills, add a vibrant array of colors, and draw Jimi's silhouette. The techniques used in this chapter are the first steps in learning to create assets within Flash, and they are commonly used for animation, illustration, game development, application design, and virtually anything else you'll do in Flash.

## Drawing Your First Box

Drawing a box is easy, so in this project you'll also look at a few ways to enhance your drawing with color and stroke properties. If you're new to Flash, fear not—you'll feel comfortable in no time.

### In this chapter

Drawing Your First Box

Coloring Fills and Strokes

Merging and Stacking Shapes

Creating Reusable Graphics

When you launch Flash for the first time, the Start screen appears, as shown in Figure 1-1. This is merely a handy launch utility, and if you find it intrusive, you can disable it in the lower-left corner of the panel. One of the reasons you may want to use the panel is that it keeps track of the last 10 documents you opened (although you can also access this list via the File→Open Recent menu command).

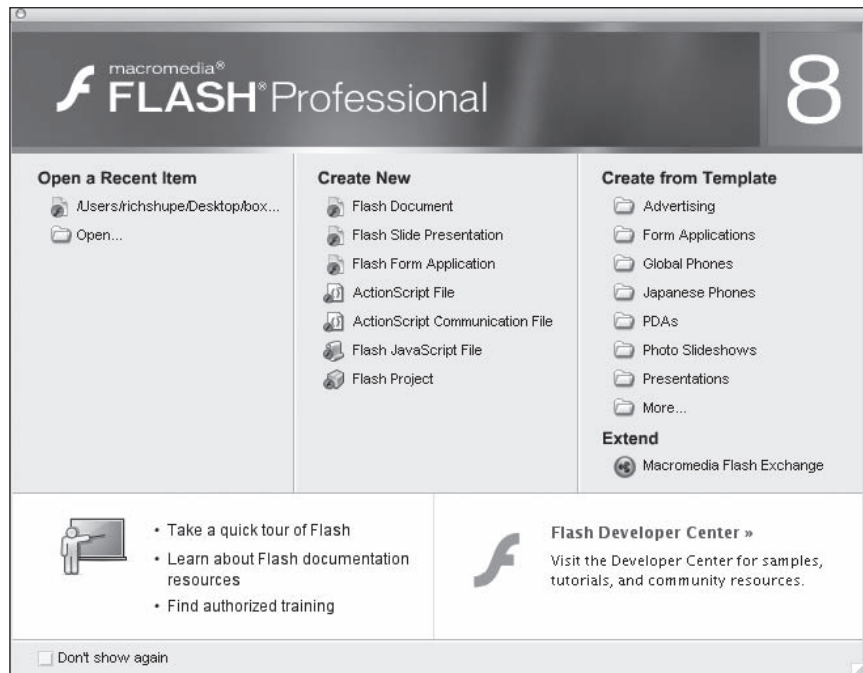


Figure 1-1. The Start screen knows what you've been doing

## Exercise Files and Trial Software

All of the exercise files in this book are available on the accompanying CD-ROM and from <http://www.flash8projects.com>. Copy or download the examples, and put them somewhere on your hard drive where you can find them easily. Hereafter, this will be called your “working directory.”

The assets you need for each exercise (images, sound files, and so on) have been grouped into numbered folders for your convenience. For example, any assets required for Chapter 1 can be found in the 01 folder. It's a good

idea to preserve this directory structure to make it easier to find any referenced files.

In many cases, completed exercise files have also been included. Occasionally, you may want to start with a specific instruction and continue from there. However, you'll benefit most by going through the exercises step by step.

If you haven't already installed Flash 8, a 30-day trial version is also included on the accompanying CD-ROM. Installing the Professional version of Flash 8 will allow you to get the most out of the exercises in this book.

The second column of the Start screen allows you to create a new Flash document. Figure 1-1 shows the Start screen from Flash 8 Professional, which includes additional file types. You'll learn a little more about some of these files later in this book. In Chapter 13, you'll also learn how to use templates. For now, though, you'll focus on your first steps.

#### NOTE

*The bottom portion the Start screen provides quick links to Internet resources to help you learn Flash. This area may reflect automatic online updates provided by Macromedia, so your screen may vary slightly.*

## Setting Up the File

Before you can start designing, you need to create a new Flash file:

1. If you haven't already, copy the CD-ROM source files and folders to your hard drive. Hereafter, the location where you store these files will be called your "working directory." Also, be sure you're running Flash 8. (Some features of this book require the Professional version. See "Getting Started" in the Preface for more information.)
2. Under the Create New heading of the Start screen, or by using the File→New→Flash Document menu command, create a new Flash document. You will save this file using the *.fla* extension (sometimes pronounced "flah"). This file type is the basis for virtually every Flash project.
3. Save your new document as *box.fla* in your working directory, using the File→Save menu option. Alternatively, you can press Ctrl-S (Win) or Cmd-S (Mac) to save the file.

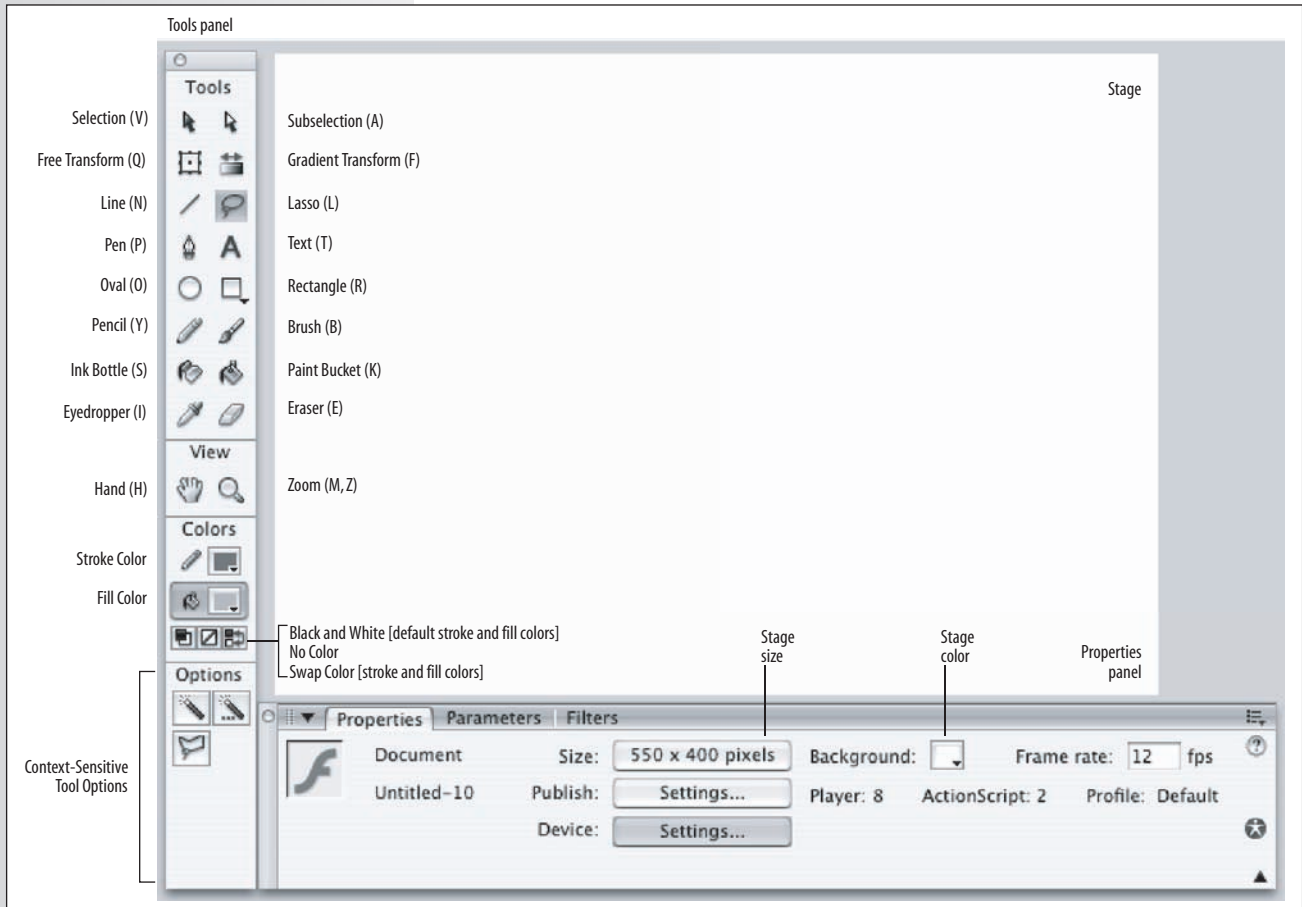
#### NOTE

*Throughout this book, exercises will frequently build upon previous exercises. Usually, you'll be told when you no longer need a file, or when to start another file for a new topic or for experimentation. So, as a new series of steps begins, assume you're continuing your work, unless otherwise stated. You will also occasionally be prompted to compare your work to the sample files at major milestones. Feel free also to check your progress against interim steps at any time.*

#### NOTE

*Flash documents are often called FLA files. This is the convention used in this book, although some developers reserve this term for the final compressed file format used to distribute Flash files. Files in this format use the *.swf* (pronounced "swiff") extension. SWF files are compiled when you test or publish your FLA files.*

Now, take a moment to look around the world of Flash. Notice the Tools panel, shown in Figure 1-2. (If your Tools panel isn't visible, open it using the Window→Tools menu command.) Many tools will look familiar if you've worked with other graphics programs. As you roll your cursor over each tool in the panel, a tool tip indicates its name and shows its keyboard shortcut in parentheses. For example, you can select the Rectangle tool by pressing R. The Options portion at the bottom of the Tools panel reflects options for the currently selected tool. This context-sensitive area changes depending on which tool is active.



**Figure 1-2.** The Flash interface, showing the Tools panel, Stage, and Properties panel

## Drawing a Rectangle

Begin by drawing a rectangle as a starting point for a new design:

1. In the Tools panel, select the Rectangle tool, indicated in Figure 1-2.
2. Roll your cursor over the Stage (the large white area in the document window, also seen in Figure 1-2). Your cursor should look like a cross-hair.

3. Click and drag somewhere on the Stage to create a rectangle. As you can see, using the Rectangle tool in Flash is similar to using the same tool in other programs.

## Using Undo and the History Panel

You are probably familiar with the Undo command from your experience with other programs. When you use the Edit→Undo menu command, or press Ctrl-Z (Win) or Cmd-Z (Mac), Flash will attempt to undo your last action. (Some things can't be undone; in this case, Flash will try to warn you before proceeding.)

Take a moment now to undo your rectangle creation. If you followed the steps described previously, you should now have an empty Stage. If not, continue to use the Undo command until your Stage is empty.

By default, Flash will store the 100 most recent undoable changes you have made to your open documents. Several hundred levels of undo can be stored, but they require memory. With each additional step you store, more system memory is consumed. Refer to Appendix A for information on how to change this setting.

You can see the undo list in action by opening the History panel, shown in Figure 1-3, using the Window→Other Panels→History menu command. Once you have opened this panel, draw a few boxes on the Stage again, and watch the history list grow. After adding a few items, use the draggable arrow on the left side of the panel to slide up the list. As you move further back in history, you will see your boxes disappear as your actions are undone. Great, huh? It gets better. In Chapter 2, you'll learn how to use the History panel to help automate tasks.

Flash 8 now offers both *Document Undo* and *Object Undo* modes. Document-level undo maintains a single list of all undoable actions for the entire Flash document. Object-level undo maintains separate lists of your actions for each major object (such as buttons and similar items) in your document. Object-level undo is more flexible, because you can undo an action in one object without having to also undo more recent actions in other objects.

However, it can sometimes be advantageous to use document-level undo, and you should bear in mind that switching between the two modes will clear your current history. (Your document will not be affected, but your history list will be emptied.) So, experiment to see which you prefer, and be cautious when switching undo modes.

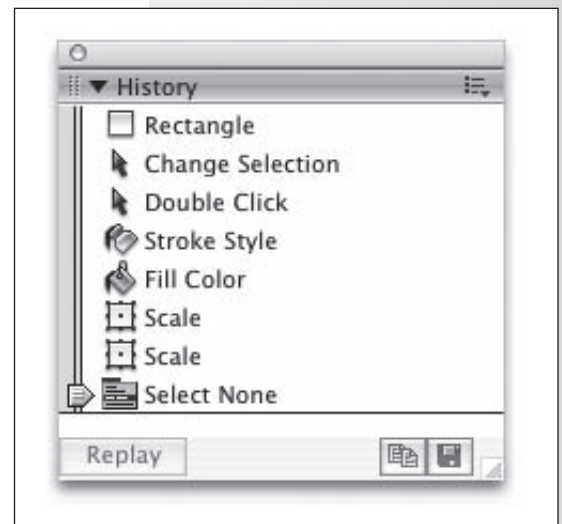


Figure 1-3. The History panel

## Drawing a Perfect Square

Drawing a perfect square is simpler than it sounds. All you need to do is hold down the Shift key while drawing the square. This technique, common to many graphics applications, is called “constraining.” This is because the Shift key constrains the shape into a square while it’s being drawn, even if you accidentally describe a rectangle with your mouse movement. This is a great way to ensure that your squares are actually square. The same rule applies to the Oval tool for drawing perfect circles and the Line tool for drawing straight lines.

What if you want to create a box that is exactly 300 × 300 pixels? In this case, you can use numeric input to get the exact shape you want. Here’s how:

1. If you haven’t already done so, delete everything on the Stage.
2. Select the Rectangle tool, and click and drag on the Stage to draw a rectangle.
3. When you’re done, choose the Selection tool (shown in Figure 1-2) from the Tools panel.
4. Double-click inside the box you just drew to select both the box and its border.

### NOTE

*Flash allows you to select individual fills and strokes separately, so be sure you select both when desired. See the “Fills and Strokes” sidebar for more information.*

5. With the box selected, locate the Properties panel, shown in Figure 1-4. (If necessary, open it with the menu command Window→Properties.) The Properties panel looks different in this figure than it did in Figure 1-2 because its contents change depending on what is selected. At the bottom of Figure 1-2, the Properties panel displays the Stage’s properties. In Figure 1-4, it displays properties for the selected rectangle.
6. The W and H fields in the Properties panel represent the rectangle’s width and height. The relative proportions of width and height can be constrained by clicking on the lock icon to the left of these fields. In this case, this is fine, as you are drawing a square. However, if you want freedom to enter any width or height, you’ll need to click on the icon to unlock it, as pictured in Figure 1-4. Enter 300 for the width and 300 for the height. You now have a precisely sized rectangle—a perfect square.
7. Save your work.

### NOTE

*Don’t worry if you can’t get an exact number when entering property values. Flash does its best to give you the value you request, but it isn’t always easy to achieve exact sizing or positioning at the sub-pixel level. If you get values like 299.9 or 300.1, the difference will usually be unnoticeable.*

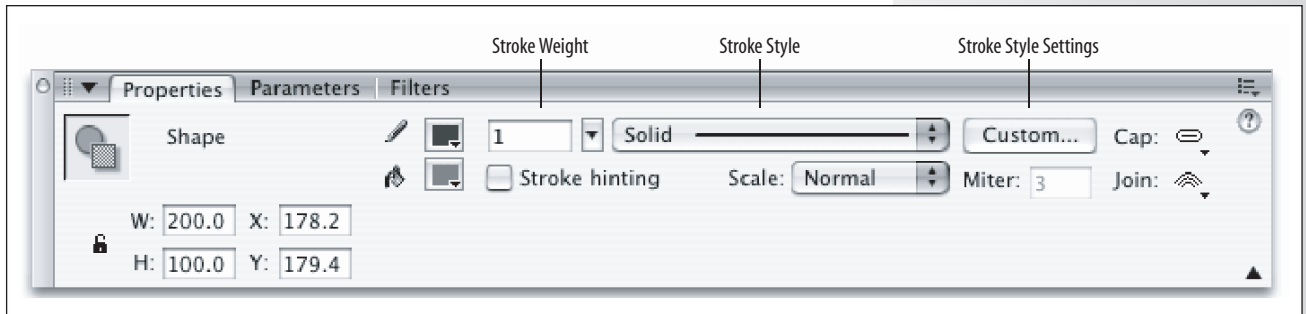


Figure 1-4. The Properties panel, as it appears when a shape is selected

## Adjusting Caps and Joins

You may have noticed, during your experimenting, that you haven't been getting nice sharp corners. Flash's default behavior uses rounded stroke corners and end caps, but Flash 8 finally introduced the ability to customize the cap and join settings of your strokes. The *Cap* setting dictates the appearance of the ends of the lines, while the *Join* setting defines the appearance of corners. As you can see in Figure 1-4, both settings can be found in the far right of the Properties panel when a shape with strokes is the active selection.

Caps can be set to *None* (the line ends precisely at the end point), *Round* (a round cap is placed on each end of the line), or *Square* (a square cap is placed on each end of the line). Joins can be set to *Miter* (sharp corner), *Round*, (rounded corner), or *Bevel* (two-point corner).

Figure 1-5 shows the results of each setting. All the line widths are the same, and all shape end points terminate at the dotted lines. From top to bottom, the features used are: 1) Cap: *None*, Join: *Miter*; 2) Cap: *Round*, Join: *Round*; and 3) Cap: *Square*, Join: *Bevel*.

### NOTE

Cap settings add to the apparent length of a line. Notice in Figure 1-5 that although all the lines end at the same horizontal or vertical positions, those with caps appear longer.

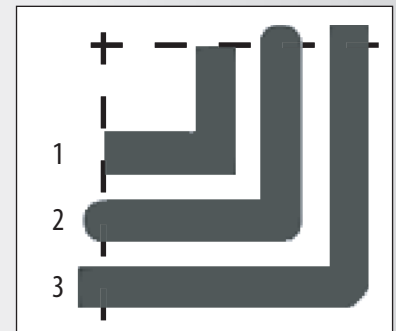


Figure 1-5. Cap and join differences

## Coloring Fills and Strokes

When you're drawing or editing a shape, Flash independently colors fills and strokes. You can change the colors via the Tools panel or the Properties panel when a shape or shape tool is active. In both locations, a small pencil icon indicates the stroke color, and a small paint bucket icon indicates the fill color.

There are two simple ways to change a shape's fill or stroke color. First, you can change the fill color of a selected shape using the pop-up color swatch utility:

1. Choose the Selection tool from the Tools panel, and click once inside the box on the Stage. The fill area will appear as if seen through a screen, indicating that it is selected.
2. In the Properties or Tools panel, click on the Fill Color swatch. A palette of colors will pop up, and you'll be allowed to select from the default web-safe swatches. A white box with a red line through it at the top of the pop-up indicates that no fill will be used, allowing underlying assets to be visible within the borders of the shape.
3. Deselect the box to view the results. You can deselect by using Edit→Deselect All, or by clicking in any unoccupied area of the Stage or Work Area (the area surrounding the Stage). Deselecting is often required to avoid editing an active element, and it will soon become second nature.

## Fills and Strokes

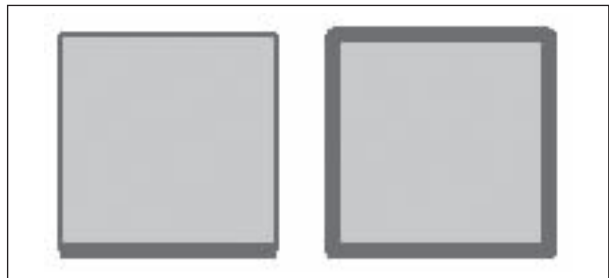
Unlike in many other drawing programs, Flash elements are drawn using separate *fills* and *strokes*. Although this can take some getting used to, it can also be very helpful. The best way to understand how fills and strokes work is to do a little experimenting. You may want to open a new document so that you can tinker while reading without affecting your ongoing exercises.

A *fill* is the area of color that fills the inside of a stroked shape or constitutes the body of a borderless shape. For example, if you draw a free-form blob on the Stage, the colored area inside its border is its fill. The colored area is called a fill even if no border surrounds it.

A *stroke* is the border or line surrounding a shape. Strokes can have six styles, the most common of which are solid, dotted, and dashed. Each stroke type can be customized with a variety of settings, all of which can be found within the Properties panel when a line or line-making tool is active.

Flash allows you to select individual fills and strokes separately, if you desire. You can select only a fill, or even only one side of a multi-sided stroke. Selecting a contiguous fill, or a single part of a multi-part stroke, requires one click. To select an entire contiguous stroke, double-click

on the stroke only. To select both a fill and its stroke, double-click on the fill. See Figure 1-6 for a comparison of single-clicking and double-clicking a multi-part stroke.



**Figure 1-6.** Stroke selection by single-click (left) and double-click (right)

If you want to work with an entire shape (fill and stroke) but accidentally only select one, the remainder of the shape will not be edited. This can happen when dragging—for example, you may drag a fill and leave the stroke behind.

If this happens, undo your change and double-click, or press the Shift key while selecting multiple objects, to include all the desired elements in your efforts.

You selected just the fill to practice, and to show that the colors of selected shapes can be changed directly with the color swatches. Next, you'll use the Ink Bottle tool to exclusively change the color of a stroke, regardless of whether or not it has been selected. The Ink Bottle tool is adjacent to the Paint Bucket tool in the Tools palette.

You may be familiar with the Paint Bucket from experience with other graphics tools. You can use it to change a fill color by single-clicking within the fill area. Think of the Ink Bottle tool as a Paint Bucket for lines. It will apply your preferred stroke attributes to any shape stroke you click on:

1. Choose the Ink Bottle tool, click on the Stroke Color chip (indicated by the pencil icon), and pick a color for your stroke.
2. If you wish, experiment with some of the other stroke attributes in the Properties panel, such as thickness, cap, and join settings. Editing strokes will be discussed further in a few minutes.
3. To view the results, with the Ink Bottle tool active, click on a shape stroke. The stroke should change to reflect your chosen attributes while the fill remains unaffected.

## Solid Fills and Gradients

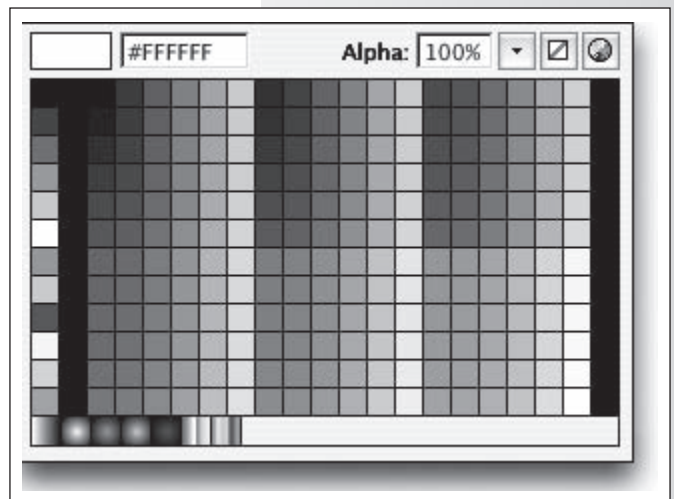
Now that you know how to add colors to shapes, you're probably asking yourself, "What if I don't want to use one of the default colors available in these tool swatches?" No problem. There are a few ways to create your own custom colors.

If you just need a quick color, which you may not use often, you can create a color on the fly, you can take advantage of two options in the color swatches pop-up palette that appears when you select a stroke or fill color. The first option is to use the hexadecimal entry field in the top left of the swatches pop-up. This field allows you to enter hex color values to specify any color numerically. If you are familiar with HTML, you will likely be comfortable with this method.

The second option is to use the color picker icon in the upper right of the swatches pop-up. This will bring up your operating system's color picker, which will allow you to select colors using a variety of methods used by your system. If you work with color in other applications, you may be familiar with this utility already.

Another handy feature in Flash 8's swatches pop-up (Figure 1-7) is the ability to control your color's opacity via the Alpha setting. Colors with an alpha value of less than 100% have some transparency, allowing your colors to mix on the Stage.

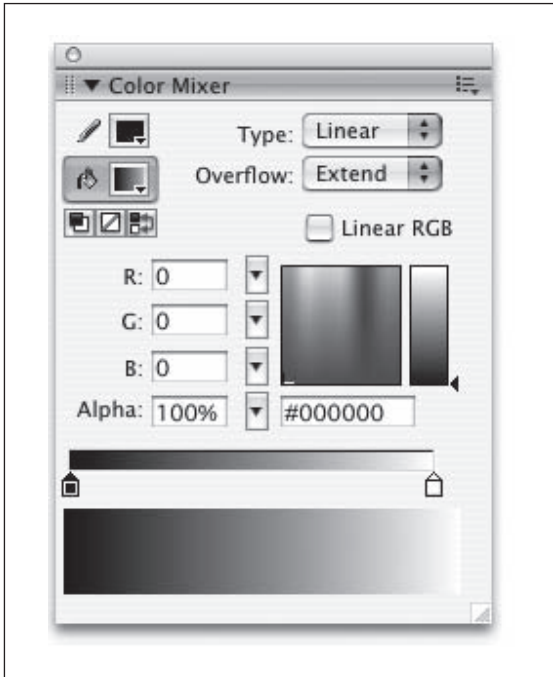
**Figure 1-7.** The color swatches pop-up, accessible via the Tools and Properties panels



The swatch pop-up utility is great for quick access, but what if you need more control? Enter the Color Mixer.

## The Color Mixer

The Color Mixer panel, shown in Figure 1-8, gives you some great options for changing strokes and fills, and even enables you to change the fill completely by converting it to a gradient.



**Figure 1-8.** The Color Mixer panel

Here is a breakdown of the options for the Color Mixer:

### Stroke color

Changes the color of the stroke. You can access your operating system's native color picker using the icon in the upper right of the pop-up palette. This allows you to specify colors from spaces other than RGB, such as hue, saturation, and brightness (HSB).

### Fill color

Changes the color of the fill. See Stroke Color for additional information.

### Swatch utilities

Sets the stroke and fill colors to black and white,

respectively; removes the selected stroke or fill; switches the stroke and fill colors.

### Fill Type (menu)

Changes the fill style. Options include None (which deletes the fill), Solid (for a solid, one-color fill), Linear (for a gradient that blends on a linear path), Radial (for a gradient that blends outward, in a circular path, from a focal point), and Bitmap (for a fill made of tiled bitmap images). Additional gradient options are discussed in the main text.

### Gradient Overflow (menu)

Visible only when a gradient fill type is selected; determines how the gradient is mapped when exceeding the range of the Gradient Transform tool. Options include Extend, Reflect, and Repeat.

### RGB input (three separate entry fields, pop-up sliders)

Changes the density of the red, green, and blue (RGB) colors in a fill.

### Alpha

Sets the percentage of opacity for a solid fill (or the currently selected Color Proxy slider for a gradient fill). Zero percent alpha makes it invisible, while 100% alpha makes it opaque.

### Color chooser

Provides a quick, visual way to choose any color. Simply click on the color chooser and drag the cursor around to locate the color you want.

### Brightness slider

Changes the brightness of the current color.

### Hexadecimal input

Changes the color by accepting hexadecimal values. Commonly used in HTML, hex values are six-digit number and/or letter (A–F) combinations that represent colors, such as #000000, which represents black, or #FFFFFF, which represents white.

### Gradient color swatches

Changes the colors used in gradients. Between 2 and 15 colors can be used for each gradient.

### Current color

Displays the currently selected color.

When creating *solid* colors, like the colors you’ve been using so far, in some ways the Color Mixer is not unlike the color swatches pop-up. However, instead of limiting custom color creation to hexadecimal value entry and the system color picker, it has its own color chooser. A small box presents a graduated tone of hues from left to right, and decreasing brightness from top to bottom. You can pick a color value with your mouse and then use the vertical slider to increase or decrease brightness. As you’ll see in a moment, you can also save your color choice for later reuse. See the sidebar “The Color Mixer” for a look at its interface and a little more detail on its functionality.

## Gradient Psychodelia

When creating *gradients*, the Color Mixer is required. Next, you’ll use it to create a gradient that will be the basis for a psychedelic look in a 1960s-style Jimi Hendrix poster illustration:

1. If you’ve been experimenting, undo your changes or use the File→Revert menu option. As long as you haven’t saved since last instructed to do so, this will revert your file back to where you left off—a 300 × 300 pixel square with fill and stroke.
2. Select the fill and choose Window→Color Mixer to open the panel shown in Figure 1-8, in the sidebar “The Color Mixer.”
3. In the Color Mixer panel, select Linear from the Type drop-down menu. The solid color automatically changes to a linear gradient. A thin bar, showing the gradient in progress, allows you to edit the gradient via the mini-swatches below it. (A new menu, called *Overflow*, also appears; you’ll learn about that later.)

The small color chips beneath the gradient color bar have arrows pointing to the focus point of each color. These swatches define the colors used in the gradient and slide left and right to define the position of each color.

To change the gradient, change the color of these swatches in one of two ways. Double-clicking a chip enables you to choose a color from the familiar swatch pop-up utility. For more flexibility, you can select a chip with a single click and enter RGB or hexadecimal values in the input fields, or use your mouse with the color picker and brightness slider. Both techniques allow you to add an alpha value to a color to achieve transparencies of 0 to 100%.

You can use from 2 to 15 swatches, and you can remove swatches by dragging them down and away from the gradient preview bar. Change the gradient hues now, and add two more for big splashes of color:

1. Double-click on the leftmost gradient color swatch, and pick a bright yellow color from the color palette. If you want to try to match the sample files (although there’s no reason you must), the yellow is HEX #FFFF00, or RGB 255, 255, 0.

### NOTE

Be sure to hit Enter/Return to update the color each time you put in a value.

2. Next, double-click on the rightmost gradient color swatch to change the end color for the gradient. Choose a vibrant purple. The sample file uses HEX #660099, or RGB 102, 0, 153.
3. Now add a new chip by clicking directly beneath the gradient color bar, approximately one third of the way along the span of the gradient, from the left. It will assume the color immediately above it. Change its color to HEX #FF0000, or RGB 255, 0, 0, for bright red.
4. Add another color halfway between chips two and four, or approximately one third of the way along the span of the gradient, from the right. This time enter a HEX value of #3399FF. You now have a colorful linear gradient.
5. Experiment by dragging the chips left and right to give each color more or less prominence. When you're done, place the first chip at the far left, chips two and three at one-third and two-thirds from the left, and the last chip slightly short of far right. This will give purple a bit more weight and will help demonstrate another feature in a minute or so.
6. To create a background more like the psychedelic posters from the 1960s, change the Type drop-down menu choice to Radial. This changes the gradient so it blends outward from a central point instead of along a linear path.
7. If you're satisfied with your gradient, save it for later use. If it's not already visible, choose Window→Color Swatches to view the Color Swatches panel. When you move your mouse over the empty white area, the pointer will change into a paint bucket. Click the mouse and, since the currently active new color is a gradient, the gradient will appear in the bottommost row of swatches. Now you can use the gradient anytime later in this file, even if you create a new gradient in the meantime.
8. Save your work and leave this file open.

## Custom Stroke Styles

The stroke for this image could be improved to better fit with the illustration style, so change it from a solid line to something funkier:

1. Deselect the fill of the box, and double-click the stroke to select the entire stroke.
2. In the Properties panel, give the stroke a dark maroon color.
3. Click on the Stroke Style drop-down list, shown in Figure 1-4, and choose the next-to-last style. (The styles aren't named in the menu, but it looks a bit like splattered paint.)
4. Using the Stroke Height text field or slider in the Properties panel, set the stroke weight to 10.

### NOTE

*Flash 8 now supports gradients as stroke colors. If you want your image to be really colorful, you can try to make a new gradient for the border. Don't worry too much if your file differs from the sample. Think outside the box!*

5. Deselect the stroke to see what it looks like. Your stroke should be 10 pixels wide and appear as an array of spots that resemble paint splatters. Things are looking good now, but the stroke doesn't seal off the border as well as it could, so modify it to fix the problem.
6. Select the entire stroke again. In the Properties panel, click the Custom button to open the Stroke Style dialog box. The Type drop-down should already be set to Stipple (unless you chose a different type of stroke earlier).
7. Set Dot Size to Medium, Dot Variation to Varied Sizes, and Density to Very Dense. This tightens up the stroke to seal up the edges of the fill it surrounds. Click OK and deselect the stroke to see your work by clicking somewhere outside of the box. Nice, huh?
8. Save the file, but leave it open.

## Scaling Shapes

Often, you can't tell when starting a drawing how big or small it needs to be, and sometimes creative tangents bring about change. Here, you need to make your illustration more impressive by filling up the Stage.

Quick, inexact scaling is very simple in Flash. All you need to do is activate the Free Transform tool (Q) and use your mouse to drag one or more of the active drag locations. You can resize by dragging any of the handles; the cursor will temporarily change to a double-headed arrow to indicate that scaling is possible. (You can also skew by dragging the sides, or rotate by dragging *outside* one of the corner handles. Temporary cursor changes will also indicate that these features are possible.)

Here, however, you need exact resizing and positioning capabilities, so you'll use the Properties panel. The Stage is rectangular, but the image is square, so you have to make one change first. If you resize the background box as is, changing its shape from a square to a rectangle, the radial gradient fill will distort. You'll learn later how to position and scale gradients directly, but in this exercise you'll delete the fill and replace it after resizing the box. Here's how:

1. First, make sure your background gradient has been saved in the Color Swatches panel, as described earlier. If not, use the Eyedropper tool to pick up the gradient from the background and then click anywhere in the empty white area of the Color Swatches panel.
2. Now you can safely select the gradient fill and delete it using the Backspace or Delete key on your keyboard.
3. Select the box's stroke and, using the Properties panel, set the width (W) to 550 and the height (H) to 400. Remember, you may need to unlock the padlock icon to change the box's width and height independently. The stroke now matches the dimensions of the Stage.

4. Also in the Properties panel, change the box's X (horizontal) and Y (vertical) coordinates to 0. After pressing Tab or Return to register the values, this aligns the stroke to the upper-left corner of the Stage.
5. Activate the Paint Bucket tool and switch the fill color using the pop-up swatches. You should now see your gradient in the bottom row with the default gradients. Select it.
6. Look in the Paint Bucket tool's Options area, located in the lower portion of the Tools panel, and make sure the Lock Fill option is disabled. Use the Paint Bucket tool to apply the gradient to the background box. Notice that the radial gradient's center point is determined by the position of your click. Click several times inside the box to change the center point for the gradient until you are happy with the look.
7. Next, activate the Lock Fill option and try to apply the fill again. The Lock Fill option prevents the fill from being changed. Now that you've seen the effect of this option in both states, disable it once again, so you can edit the fill later. If the scale or position of the gradient isn't exactly the way you want it, don't worry. You'll soon learn how to change that.
8. Save your work. If you want to compare your work with the sample files, your file should look like *box\_03.fla* in the *01* folder in your working directory.

## Drawing Strokes Freehand

Adding a silhouette of Jimi Hendrix in the foreground of this gradient will finish off the design nicely. If your drawing skills aren't great, don't sweat it. The silhouette of Jimi, just like the rest of this image, can be as psychedelic as you want. The objective is to see how the Pencil tool can be used to draw shapes freehand:

1. Choose the Pencil tool from the Tools panel. In the lower portion of the panel, the context-sensitive Options area includes two options. You'll use the Object Drawing option in a moment. For now, this should be disabled. The Pencil Mode option offers you a choice of Straighten, Smooth, and Ink. These options will help you draw straighter lines, create smoother curves, or work freehand, respectively. Set the option to Smooth. In the Properties panel, set the stroke weight to 5, set the stroke color to black, and change the stroke style back to a solid line.
2. Click and drag inside the box to draw a silhouette of Jimi, like the one in Figure 1-9. Don't let the ends of the stroke touch the border of the box. (You'll learn why in a moment.)



**Figure 1-9.** A silhouette of Jimi Hendrix drawn with the Pencil tool

#### NOTE

*If you prefer not to draw the silhouette yourself, open `hendrix_silhouette.fla`, located in the 01 folder in your working directory. Double-click to select the entire stroke, choose `Edit→Copy` to copy it to your clipboard, and then return to `box.fla` and choose `Edit→Paste in Center` to paste it into `box.fla`. Close `hendrix_silhouette.fla` by choosing `File→Close` when you're done.*

You can stop and start as much as you want using the Pencil tool, so feel free to use as many lines or curves as you need. If you make a mistake, either undo it or just go with it. In creative situations like these, mistakes can sometimes be as inspirational as the lines you draw intentionally.

If you're not satisfied with your drawing, try smoothing the stroke with the Selection tool's Smooth option in the Options section of the Tools panel. This is similar to the Pencil tool's Smooth option, but it allows you to smooth strokes after they've been created. Try this:

1. Select the stroke you wish to smooth. If the lines are all connected, double-clicking will select them all. If not, press Shift and select each line individually until you have selected all of them.
2. Click once on the Smooth option at the bottom of the Tools panel. This makes the stroke smoother. If you're happy with the new look, leave it alone. If not, click Smooth one or two more times until you're satisfied. The trick is knowing when to stop smoothing it. Making it too smooth

will diminish the hand-drawn quality. Deselect the stroke when you're done.

3. Choose the Paint Bucket tool, and change the fill color to black using the Properties panel. With the Paint Bucket tool active, click once inside the outline to fill in the silhouette of Jimi.

Did the entire image turn black? If it didn't, that's fine. In fact, this is the desired result, as it shows that when you drew your line you fully enclosed the area within it (either by connecting the stroke's end point with its starting point, or by connecting both ends of your stroke with the border of the box).

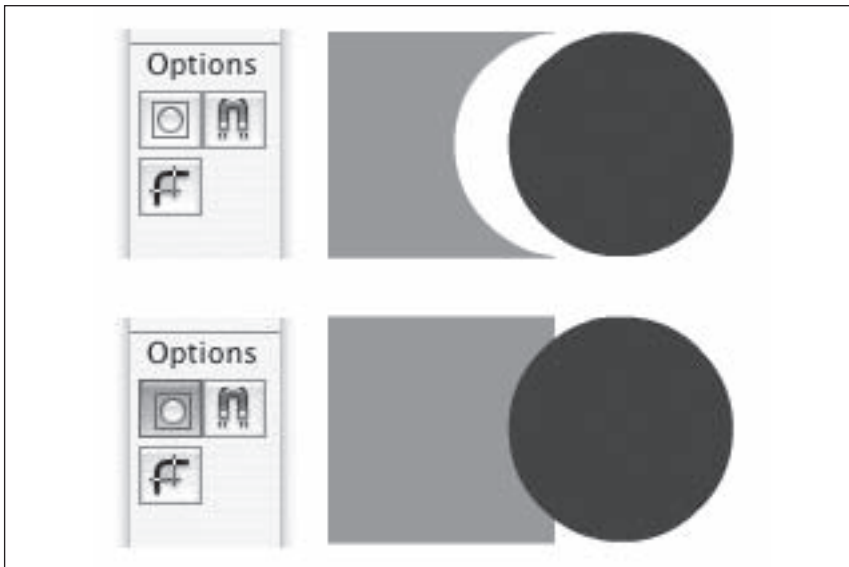
If the entire image did turn black, don't worry. What this means is that when you drew the silhouette, you didn't fully complete the Hendrix outline—a gap remained that allowed the fill color to “spill out” and fill any connected area with the black color. You can fix this by closing the stroke and sealing off the fill area:

1. Undo the last step to remove the black fill from the image.
2. Using the Pencil tool, make sure the entire outline is complete, including a line across the bottom of the silhouette.
3. Activate the Paint Bucket tool again, and display the Gap Size menu from the Options section of the Tools panel. The Gap Size option dictates how large a gap Flash will bridge for you when applying a fill with the Paint Bucket tool. Choose Close Small Gaps from the menu and click inside the shape to see if the fill is applied properly. If not, choose Close Medium Gaps and try again. If that doesn't work, choose Close Large Gaps, and the fill should be applied.
4. Save your work.

If you connected your silhouette border to the box stroke, your results may differ slightly from the descriptions in some upcoming exercises. Feel free to continue with that in mind, or open *box\_05.fla* to continue.

## Merging and Stacking Shapes

So far, you've been using Flash's default *Merge Drawing mode*. In this mode shapes can interact with one another, merging together when they overlap. For example, if the corners of two squares overlap, the frontmost shape will “knock out” the underlying shape, effectively deleting it, as seen in the top of Figure 1-10. This behavior contrasts with that of the newly introduced Object Drawing mode, which will be discussed shortly.



**Figure 1-10.** Merge Drawing mode (top) and Object Drawing mode (bottom) compared

What are the implications of shapes merging? If you didn't save your work at the end of the last sequence, do so now, and then try this with your current file:

1. Activate the Selection tool, select the black fill you just applied to the silhouette, and then delete it. Everything inside the fill area is removed, including the gradient previously behind the silhouette's fill!
2. Undo the last step. You'll need the silhouette fill for something else in a minute.

At first you may think this is a problem, but it can be very useful, *and* you can prevent it from happening, if you wish. First, however, look at a few simple examples of how Merge Drawing mode can be advantageous. If you recently saved your work, feel free to experiment.

Select the Line tool and draw a line all the way through your artwork, from outside left to outside right. (This is arbitrary, but ensures that the line goes completely through the box and protrudes on either side.) Now choose the Selection tool and click once in various regions of your art. You'll see that the line has effectively segmented your art, allowing easy selection of smaller pieces.

Next, select the Brush tool. This will show you a good example of how unprotected shapes can be altered in creative ways. In the tool's Options area, experiment with the different Paint modes. Paint Normal will behave the way you likely expect. Paint Fills, however, will paint only fills, leaving strokes unaffected. Try Paint Behind and Paint Inside to see how they work, too.

Finally, try one last thing. Select the Lasso tool, and freely select any combination of fill and stroke. With a selection active, you can move it, or delete it, as before. The Merge Drawing mode and the Lasso tool make selecting partial fills and strokes very easy and natural.

## Groups

When you don't want shapes to merge together, there are a few ways to prevent this from happening. In just a few minutes, you'll learn how to convert shapes into another type of Flash asset. Also, in Chapter 3 you'll learn how to add new layers to prevent shapes from ever touching each other. However, you still need to know how to protect existing shapes in the same layer. The first approach you'll practice is making a *group* from a shape:

1. Select the fill and outline (stroke) of the Hendrix silhouette. Press Ctrl-G (Win) or Cmd-G (Mac). This creates a group of the fill and stroke so they can easily be moved together (i.e., they cannot be edited separately). A light blue bounding rectangle appears around the silhouette when it's selected, indicating that this is no longer a shape. In this case, it is now a group.
2. Drag the group to the right. You'll see that the area under the silhouette is still empty, but if you deselect the silhouette and then move it again, you will find that this time the move does not affect the gradient beneath it.
3. Save your work. If you wish, compare your file to *box\_06.fla*.

Traditionally, you probably think of a group as two or more items treated as a single entity, and the same is certainly true in Flash. However, as you've just seen, a group in Flash has another important purpose. By grouping the parts of a shape, you ensure that the shape can no longer be altered by another shape. Thus, grouping single items is an attractive solution to unwanted shape merging.

## Editing groups

What if you want to edit a shape that has been grouped? Although the shapes no longer seem editable, there are two ways to satisfy this need.

First, if you double-click a group, the surrounding assets will appear to dim, and the group will become editable. This is because you are effectively "inside" the group and can edit it as if it were a shape. When you are finished editing, you can double-click elsewhere on the Stage to go back "outside" the group. This is useful when you want the shape to continue to be protected from interaction with other shapes.