Team LiB

The Hidden Power of Flash Components
by J. Scott Hamlin, Jared Tarbell and Brandon Williams
ISBN:0782142109
Sybex © 2003 (304 pages)

Use this step-by-step guide to achieve amazing effects with the hundreds of Flash components available from Macromedia and various third parties, from avoiding pitfalls to building games and more.

Table of Contents

Team LiB
Team LiB

Back Cover

You don't have to be a programmer to achieve amazing effects with the hundreds of Flash components available from Macromedia and various third parties. Step by step, *The Hidden Power of Flash Components* shows you how.

For those of you with just a little programming experience, this book also demonstrates how to build your own components to use and share with other developers. Whatever your aims, and regardless of your experience, soon all the power of this incredible Flash feature will be within your reach. Coverage includes:

- Determining the most effective ways to leverage components
- Avoiding component pitfalls
- Customizing external resources for use with components
- Building your own components
- Building a custom UI, Live Preview, and MXP file for a component
- Customizing component artwork

- Using components to build games
- Understanding the difference between components and SmartClips
- Troubleshooting component construction and application
- Using multiple components to create more complex effects

Team LiB

The Hidden Power of Flash Components

J. SCOTT HAMLIN WITH
JARED TARBELL |
BRANDON WILLIAMS

SAN FRANCISCO | LONDON
Associate Publisher: DAN BRODNITZ

Acquisitions Editor: MARIANN BARSOLO

Developmental Editor: PETE GAUGHAN

Production Editor: LESLIE E.H. LIGHT

Technical Editor: DANIEL GRAY

Copyeditor: ANAMARY EHLEN

Compositor: KATE KAMINSKI, HAPPENSTANCE TYPE-O-RAMA

CD Coordinator: DAN MUMMERT

CD Technician: KEVIN LY

Proofreaders: EMILY HSUAN AND SARAH TANNEHILL

Indexer: TED LAUX

Interior Designer: CARYL GORSKA

Cover Designer: INGALLS + ASSOCIATES

Cover Illustrator/Photographer: ØIVIND SANDUM

Copyright © 2003 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501.

World rights reserved. The author(s) created reusable code in this publication expressly for reuse by readers. Sybex grants readers limited permission to reuse the code found in this publication or its accompanying CD-ROM so long as the author(s) are attributed in any application containing the reusable code and the code itself is never distributed, posted online by electronic transmission, sold, or commercially exploited as a stand-alone product. Aside from this specific exception concerning reusable code, no part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic, or other record, without the prior agreement and written permission of the publisher.

LIBRARY OF CONGRESS CARD NUMBER: 2003101648

ISBN: 0-7821-4210-9

SYBEX and the SYBEX logo are either registered trademarks or trademarks of SYBEX Inc. in the United States and/or other countries.

Savvy is a trademark of SYBEX Inc.

Screen reproductions produced with FullShot 99. FullShot 99 © 1991-1999 Inbit Incorporated.

All rights reserved.

FullShot is a trademark of Inbit Incorporated.

The CD interface was created using Macromedia Director, COPYRIGHT 1994, 1997-1999 Macromedia Inc. For more information on Macromedia and Macromedia Director, visit http://www.macromedia.com.

TRADEMARKS: SYBEX has attempted throughout this book to distinguish proprietary trademarks from descriptive terms by following the capitalization style used by the manufacturer.

The author and publisher have made their best efforts to prepare this book, and the content is based upon final release software whenever possible. Portions of the manuscript may be based upon pre-release versions supplied by software manufacturer(s). The author and the publisher make no representation or warranties of any kind with regard to the completeness or accuracy of the contents herein and accept no liability of any kind including but not limited to performance, merchantability, fitness for any particular purpose, or any losses or damages of any kind caused or alleged to be caused directly or indirectly from this book.

MANUFACTURED IN THE UNITED STATES OF AMERICA

10 9 8 7 6 5 4 3 2 1

Dear Reader,

Thank you for choosing *The Hidden Power of Flash Components*. This book is part of a new wave of Sybex graphics books, all written by outstanding authors-artists and professional teachers who really know their stuff, and have a clear vision of the audience they're writing for.

At Sybex, we're committed to producing a full line of quality digital imaging books. With each title, we're working hard to set a new standard for the industry. From the paper we print on, to the designers we work with, to the visual examples our authors provide, our goal is to bring you the best graphics and digital photography books available.

I hope you see all that reflected in these pages. I'd be very interested in hearing your feedback on how we're doing. To let us know what you think about this, or any other Sybex book, please visit us at www.sybex.com. Once there, go to the product page, click on Submit a Review, and fill out the questionnaire. Your input is greatly appreciated.

Best regards,

Daniel A. Brodnitz
Associate Publisher-Graphics
Sybex Inc.

Software License Agreement: Terms and Conditions

The media and/or any online materials accompanying this book that are available now or in the future contain programs and/or text files (the "Software") to be used in connection with the book. SYBEX hereby grants to you a license to use the Software, subject to the terms that follow. Your purchase, acceptance, or use of the Software will constitute your acceptance of such terms. &"Owner(s)"). You are hereby granted a single-user license to use the Software for your personal, noncommercial use only. You may not reproduce, sell, distribute, publish, circulate, or commercially exploit the Software, or any portion thereof, without the written consent of SYBEX and the specific copyright owner(s) of any component software included on this media. In the event that the Software or components include specific license requirements or end-user agreements, statements of condition, disclaimers, limitations or warranties ("End-User License"), those End-User Licenses supersede the terms and conditions herein as to that particular Software component. Your purchase, acceptance, or use of the Software will constitute your acceptance of such End-User Licenses. By purchase, use or acceptance of the Software you further agree to comply with all export laws and regulations of the United States as such laws and regulations may exist from time to time.

Reusable Code in This Book The author(s) created reusable code in this publication expressly for reuse by readers. Sybex grants readers limited permission to reuse the code found in this publication, its accompanying CD-ROM or available for download from our website so long as the author(s) are attributed in any application containing the reusable code and the code itself is never distributed, posted online by electronic transmission, sold, or commercially exploited as a stand-alone product.

Software Support Components of the supplemental Software and any offers associated with them may be supported by the specific Owner(s) of that material, but they are not supported by SYBEX. Information regarding any available support may be obtained from the Owner(s) using the information provided in the appropriate read.me files or listed elsewhere on the media. Should the manufacturer(s) or other Owner(s) cease to offer support or decline to honor any offer, SYBEX bears no responsibility. This notice concerning support for the Software is provided for your information only. SYBEX is not the agent or principal of the Owner(s), and SYBEX is in no way responsible for providing any support for the Software, nor is it liable or responsible for any support provided, or not provided, by the Owner(s).

Warranty SYBEX warrants the enclosed media to be free of physical defects for a period of ninety (90) days after purchase. The Software is not available from SYBEX in any other form or media than that enclosed herein or posted to www.sybex.com. If you discover a defect in the media during this warranty period, you may obtain a replacement of identical format at no charge by sending the defective media, postage prepaid, with proof of purchase to:

SYBEX Inc.
Product Support Department
1151 Marina Village Parkway
Alameda, CA 94501
Web: http://www.sybex.com

After the 90-day period, you can obtain replacement media of identical format by sending us the defective disk, proof of purchase, and a check or money order for $10, payable to SYBEX.

Disclaimer SYBEX makes no warranty or representation, either expressed or implied, with respect to the Software or its contents, quality, performance, merchantability, or fitness for a particular purpose. In no event will SYBEX, its distributors, or dealers be liable to you or any other party for direct, indirect, special,

incidental, consequential, or other damages arising out of the use of or inability to use the Software or its contents even if advised of the possibility of such damage. In the event that the Software includes an online update feature, SYBEX further disclaims any obligation to provide this feature for any specific duration other than the initial posting. The exclusion of implied warranties is not permitted by some states. Therefore, the above exclusion may not apply to you. This warranty provides you with specific legal rights; there may be other rights that you may have that vary from state to state. The pricing of the book with the Software by SYBEX reflects the allocation of risk and limitations on liability contained in this agreement of Terms and Conditions.

Shareware Distribution This Software may contain various programs that are distributed as shareware. Copyright laws apply to both shareware and ordinary commercial software, and the copyright Owner(s) retains all rights. If you try a shareware program and continue using it, you are expected to register it. Individual programs differ on details of trial periods, registration, and payment. Please observe the requirements stated in appropriate files.

Copy Protection The Software in whole or in part may or may not be copy-protected or encrypted. However, in all cases, reselling or redistributing these files without authorization is expressly forbidden except as specifically provided for by the Owner(s) therein.

Acknowledgments

Thanks to the team at Sybex, particularly Mariann Barsolo and Pete Gaughan for going beyond being top-notch professionals, but also being a real pleasure to work with. I'd also like to thank Brandon Williams and Jared Tarbell for bringing their impressive skills to this book, both with many of the components and several of the chapters in this book. Thanks also to Brian Finlay for his fun cartoon images in Chapter 1. Also, I'd like to thank Scott Balay for developing some of the components featured in this book. Thanks also to Daniel Gray for his technical editing. I'm also thankful for my son Aidan and my daughter Audrey, who help me keep my perspective with playful breaks. Thank you isn't enough for the support my wife, Staci, has given me as I worked on this book and throughout my so-called career. Finally, out of sheer humble recognition of the raw facts, I am, as always, compelled to acknowledge the source. It is fully by the grace of God that I came to be capable of producing this book. For my part, it is He who should be given all credit-not me.

About the Author

Scott Hamlin is the director of Eyeland Studio, Inc. (www.eyeland.com). Eyeland Studio is a web-content design studio specializing in Flash game production and original content design. Eyeland Studio's client list includes Wrigley, Scholastic, Basic Fun, Nokia, Procter & Gamble, Sun Microsystems, MTV Europe, and Nabisco. Eyeland Studio produces several products sold over the Web and on CD. Eyewire.com, one of the world's largest stock imagery companies, carries Eyeland Studio's products.

Scott is also the author/coauthor of numerous books, including *Flash MX ActionScript: The Designer's Edge* with Jennifer Hall.

Team LiB
Team LiB

Introduction

Macromedia Flash has progressed far beyond its humble roots. When Flash was initially released, the appeal was based on little more than the advantage of having resolution-independent interfaces and animations

download quickly over the Internet. Early users were wowed merely by the ability to implement interactive buttons without the need to program.

In the context of the Web at the time, that was hot stuff, and it started something of a revolution. Now Flash is all but the de facto Internet multimedia standard. As such, people of just about every discipline, from former print graphic designers and HTML programmers to teachers and lawyers, are getting interested in using Flash to bring their visions of Internet paradise to life.

Many of those new to Flash are in for a rude awakening because in order to be competitive on the increasingly sophisticated realm of the Web, Flash files frequently require significant Flash programming (ActionScript). Although ActionScript's genesis was also relatively humble, the scripting language for Flash has evolved into something that is often beyond the means of many beginners to learn in a short amount of time.

Components are the answer to this dilemma. Components allow Flash users to implement functionality that is based on advanced ActionScript without the need to actually work with the ActionScript. While components are not always a comprehensive solution to every conceivable requirement, they can, at the very least, provide significant shortcuts toward the required ends.

Components can also be very useful for advanced Flash users as well as beginners. For advanced programmers, components offer the ability to implement necessary functionality quickly and easily—allowing the programmers to focus on something else.

Who Should Read This Book?

Anyone who is even half serious about working with Flash should learn to utilize components. This book is for anyone who wants to learn how to leverage components to work more effectively and efficiently. The book introduces you to using various types of components. It also shows you how to create components. If you are completely new to components, I recommend that you read the first few chapters before moving to more advanced chapters.

If you are familiar with components and you are interested in some of the specific components featured in this book, there is no reason why you cannot skip ahead to those chapters.

Each chapter features one or more components that you'll find on the accompanying CD. I recommend that you copy these files to your local computer so that you can save any changes and so that you can test any changes that you make.

Team LiB
Team LiB

How Is This Book Organized?

This book begins by assuming that you have little or no experience with components and moves you from relatively basic material through to more advanced topics:

Chapter 1: What Are Components and Why Should I Care? This chapter explains the incredible value of using components and discusses common misconceptions concerning components. Specifically, you'll see that the components that come with Flash MX are not indicative of the real raw potential that components truly embody.

Chapter 2: Using Components 101 We jump right into the basics of using components, especially the difference between the default interface for editing component parameters and a custom interface for editing

clip parameters.

Chapter 3: Using Components with Custom User Interfaces Find out how to customize components and how you work with components in general. In particular you will learn the advantages of working with components that utilize a custom user interface.

Chapter 4: Button Components This chapter leads you through using and customizing various button components that come on the companion CD. You'll learn how to use a single component to create several different animated button effects, and also learn about components that utilize Live Preview and how to install an MXP file.

Chapter 5: Interface Components See how to utilize components to create popular interface elements using the example of a drop-down menu. You will learn how to implement navigational elements with components that use goto commands, Load Movie, and function calls.

Chapter 6: Animation Components You will see how to generate an array of animation effects. Explore the value of combining duplicates of components that have slightly different parameter settings—or even completely different art or audio resources—to enrich their effects.

Chapter 7: Text Effect Components Chapter 7 looks at components that deal with text and text-related issues. In particular, this chapter shows you how to work with shared fonts and embedded fonts.

Chapter 8: Video and Audio Components This chapter steps you through using and customizing various video and audio components provided on the book's CD. You will learn how to utilize the components to generate controls for video and audio assets, such as volume controls and fast-forward/rewind controls.

Chapter 9: Game Components You'll learn how to utilize the components to generate Flash-based games such as a concentration game and a quiz game. The components featured in this chapter contain more parameters than those featured in previous chapters; they demonstrate that components can be used to create elements that go well beyond the scope of the components that ship with Flash MX.

Chapter 10: Creating Components Move beyond just using, or customizing, premade components, and start building your own. We turn our attention to ActionScript, so you can understand all aspects of how a sample component is created.

Chapter 11: Component Extras Using a component from Chapter 10, I show you how to generate a professional-looking custom user interface for your component. You will also learn how to utilize Live Preview and how to create MXP files.

Chapter 12: Troubleshooting Components In this chapter you will learn how to deal with common troubleshooting issues related to building and testing components. You'll learn how to embed the user interface in your Flash MX components and how to make components compatible with Flash 5.

The "Hidden Power" Components

Component One of the most important parts of this book are the Flash components provided free on the CD. Not only does the CD carry two dozen components that are explained in detail in the text, but it includes many more components from several Flash developers—over 75 components in all, any one of which will save you time and make your web pages more interesting and effective.

The components featured in this book are not just simple ones used to create radio buttons. They generate rich Flash content such as dynamic animations, navigational systems, and games. However, even components that would be considered "basic" can prove to be very valuable, so, I recommend that you start with the earlier

chapters and read the book from beginning to end.

These components are meant for readers of this book only and should not be shared except as described in the CD README file. They must be installed into Flash to be accessible.

Practice Image Files

On the CD All images, movies, sound files, or animations used as practice files in the book are provided on the CD so you can work along with the exercises. They are Mac- and Windows-compatible and in common formats supported by Flash. These files are for educational purposes only and should not be used elsewhere except as described in the CD README file.

Compatible with Windows and Macintosh

Just as Macromedia Flash works on both Macintosh and Windows operating systems, the book always gives shortcuts for both so that users on either platform can successfully use the book and techniques. The standard notation for shortcuts gives Mac and Windows keys at the same time: Mac/Windows + (whatever). For example, Command/Ctrl+O will open an image, that is, use Command+O on a Mac, and Ctrl+O on a PC. The following table of keyboard equivalents will cover almost any situation.

MACINTOSH	WINDOWS	EXAMPLE
Shift	Shift	Shift+X
Option	Alt	Option/Alt+X
Command	Ctrl	Command/Ctrl+X
Control+click	right-click	Control/right-click

Team LiB
Team LiB

How to Contact the Authors

Scott Hamlin is the director of Eyeland Studio, Inc. and can be reached at <jshamlin@eyeland.com> or by visiting www.eyeland.com, www.flashfoundry.com, or www.gamesinaflash.com. Brandon Williams can be reached at <mbw234@nyu.edu>, and Jared Tarbell can be reached at <jt@levitated.net>.

Team LiB
Team LiB

Chapter 1: What are Components and Why Should I Care?

Overview

Components are valuable resources for Macromedia Flash designers, whether you are a beginner or an advanced user. There is no reason to be intimidated by components. Most components are very accessible, easy-to-use resources. In fact, typically you can leverage the value of many components with little or no Flash experience.

Dive into this chapter. I'll show you what a component is, what it can do, and how to recognize one.

- **The power of components**
- **What components are, in detail**
- **What to do with components**
- **Where to find components**

Team LiB
Team LiB

The Power of Components

Probably the quickest and easiest way to understand what components are and why they are valuable is to compare them with a product you already know. In many ways, Flash components are similar to plug-ins for programs such as Adobe Photoshop or Discreet 3ds max. Components also have similarities to macros used by popular word processors like Microsoft Word. But to call components "plug-ins" or "macros" would be a disservice. They are much more powerful than that, as suggested by Figure 1.1.

Figure 1.1: Would it be immodest of me to tell you that I was the model for this illustration?

Components are resources that have the Flash programming (ActionScript) encapsulated with any other required resources (graphics, text, audio, etc.). If you can use ActionScript to program something in Flash, you can turn that something into a component.

Components are particularly useful for automating any kind of repetitive or complex task that requires ActionScript. While we'll be looking at the value of components more from a user's perspective in this book, let's look at an example on the programming side to illustrate the usefulness of components.

Let's say you need a bunch of slick drop-down menus for a variety of Flash interfaces. You are a very good ActionScript programmer and can easily program the drop-down menu. Once you program the drop-down menu, you have useful code that you can use again and again.

While most programmers are in the practice of using their code repeatedly, it still takes a lot of time for even the quickest and most incredibly talented programmers to make modifications to their code. Often programmers only need to modify variables in their code. However, those variables might be buried in hundreds of lines of code, or the programmer might need to modify multiple references to the variables or to modify a large array of variables.

In fact, it's not uncommon for programmers to spend more time modifying their existing code than they did when they originally programmed the code. If a programmer comes back to their code six months after they originally coded it, they might easily forget a variable reference or something else important that could lead to

hours and hours of bug hunting.

This often leads programmers to simply recode the item. Whether you're a programmer or not, you can probably appreciate the fact that it is very inefficient to continually recode something from scratch.

Components help you avoid such a fate. Developers can create components so that the variables for the code can be more easily modified through a much more accessible interface. This, in turn, allows programmers and nonprogrammers alike to leverage that code repeatedly without tediously rooting through the actual code to make modifications.

As mentioned earlier, if you can generate something with Flash and ActionScript, then you can make a component out of it. You can make components generate any sort of effect that can be created using Flash and ActionScript, including dynamic menus, programmatic animation effects, and audio or video controls. You can even use components to generate entire games.

Your Flash projects can even contain multiple components. You could, for example, create a Flash site that contains a component that generates the navigation system along with a component that controls what audio is playing in the background.

If you are relatively new to Flash, components let you add advanced functionality into your projects without requiring that you learn programming. If you are an advanced Flash designer or programmer, components save you time by automating repetitive tasks. You can then put that time towards making your projects all the more compelling.

Components do not increase the download time of a Flash movie. Flash components are made from the same resources you would use if you didn't use a component. A drop-down menu made with a component usually uses the same resources as a drop-down menu created without a component. The only difference is that the component is easier to use and modify.

On the CD This book is about showing you how to leverage the incredible value of components. In the following chapters, you will learn how to use, manage, and even create components. The book's companion CD gives you all of the components featured in this book as well as many additional sample components.

When you are finished with this book, you will be able to "save the world before bedtime" and leap tall deadlines with a single bound. You'll feel compelled to think of a cool super hero name for yourself, and you'll soon be making sketches for your new costume, all the while wondering how good you'll look in spandex and tights.

Well okay, you may not find yourself hiding your secret identity with a pair of cheap glasses while fighting off a sudden crush for anyone named Lana Lang, but learning to utilize components will certainly empower you to do more things in much less time with Flash.

Team LiB
Team LiB

All Right, Already!! So What Is a Component Exactly?

A *component* is a movie clip in Flash that has been prewired to generate some sort of functionality. As you will see later in the book, you begin generating a component by creating a standard movie clip. You then transform the movie clip into a component with a few relatively simple steps.

Components were introduced in Flash version 5 as *smart clips*. The name indicated a movie clip with smarts added to it. Macromedia changed the name from smart clips to components in Flash MX. If you see

something referred to as a smart clip, you can treat it exactly as you would treat a component.

Smart clips developed in Flash 5 work perfectly well in Flash MX. Components that are programmed in Flash MX can be made to work in Flash 5. However, if you use code that was introduced in Flash MX in your components, they will not work in Flash 5.

So if a component is like a movie clip, how can you tell a component apart from a movie clip? The easiest way is by looking at its icon in the Flash Library. Figure 1.2 shows the library of a component called Link Scroller from Eyeland Studio's Flash Foundry (www.eyeland.com). The item highlighted, named "link scroller," is a component. Several other elements are visible in the library, all of which are actually used by the component and contained within the component.

Figure 1.2: You can tell components apart from movie clips, buttons, graphic symbols, bitmap symbols, and audio symbols by their icons.

Many of the components featured in this book are from Eyeland Studio's Flash Foundry product. Similar components can be found at www.eyeland.com or www.flashcomponents.com.

The default component icon is this:  . The standard movie clip icon is this:  . If you see the default component icon, then you know you're dealing with a component.

However, Macromedia has provided several icons that you can use for components and has introduced the capability to create custom component icons. Any resource that displays one of the following icons will also be a component. Fortunately, both the alternative icons and custom icons are not used very frequently. Most components simply utilize the default component icon.

See Chapter 11 for information on how to create a custom icon for your components.

Identifying a Component on the Stage

When you see an item on the stage, it can be difficult to determine if it is a component or some other type of Flash resource (movie clip, button, graphic symbol, etc.). As with any Flash resource, if you are in doubt about what kind it is, you can always select it and look at the Properties panel. For example, Figure 1.3 shows the Properties panel for a movie clip. You can tell at a glance that the object is a movie clip because of its icon and its Movie Clip label.

Figure 1.3: The Properties panel clearly indicates what type of object you have selected-in this case, a movie clip.

When a component is selected, the Properties panel can take on several appearances. Figure 1.4 shows the Properties panel for a component that comes with Flash MX. On the top left, you can see the default component icon and the Component label-both of which, of course, indicate that a component is selected. Notice the two tabs on the top right: Parameters and Properties.

Figure 1.4: The Properties panel displays several indicators when you have selected a component.
The Parameters tab is displayed in Figure 1.4; this tab is only visible when a component is selected. Also, notice the area within the Properties panel that looks like a small database field. This is probably the most noticeable indication that a component is selected: these are the parameters for the component.

This area is referred to as the *component user interface* (*UI*). In Figure 1.4, this is the default user interface. Component parameters allow you to customize a component. If you think of a component as a sort of machine or engine that can be used to create something, then the component's user interface is like the machine's control panel (see Figure 1.5). The UI allows you to adjust specific characteristics for whatever the component is making.

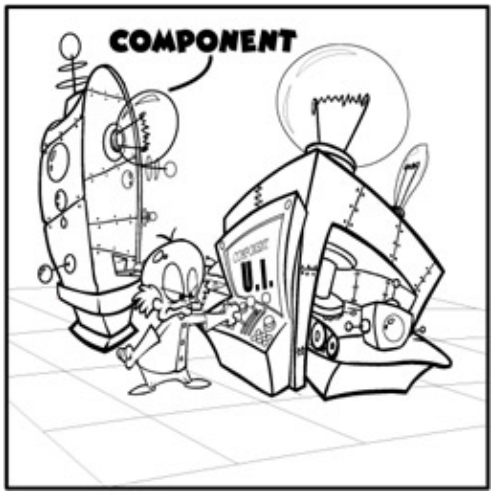

Figure 1.5: The component user interface is like a control panel on a machine.
So if a component is like a machine that makes a doohickey, then the component UI is like the machine's control panel that lets you tell the machine to make a red doohickey rather than a blue doohickey.

Figure 1.6 shows the Properties panel for a selected component with a different type of parameter user interface-in this case, a custom user interface. We will look at component UIs and parameters, including custom UIs, in detail in Chapter 2 (and, really, throughout this book).

Figure 1.6: The Properties panel for a component with a custom user interface. You must click the Launch Component Parameters Panel button to view the custom UI

Team LiB
Team LiB

What Can I Do with Components?

You can use components to generate almost anything you can imagine (anything that Flash is capable of creating, that is)—games, dynamic menu systems, interactive buttons, audio controls, video controls, transition effects, database systems, interactive mouse effects, animation effects, and much, much more. This might be a little bit of a surprise if you have noticed the components that come with Flash MX.

Figure 1.7 shows the included components. You can open this panel by selecting Components from the Windows menu. If you drag a few of these onto the stage and then publish your movie, you won't exactly receive the thrill of a lifetime. The components that come with Flash generate relatively humble things like radio buttons.

Figure 1.7: The components that come with Flash MX

If you care to inspect the code of some of these components, such as the ComboBox component, you'll see that the ActionScript used to create the component is actually somewhat complex. Nevertheless, these standard Flash MX components merely scratch the surface of what is possible with components.

To begin to appreciate the real value of components, we need to look at a few examples of components in action. Figure 1.8 shows an interface that uses two different components. One component generates the navigation system on the left, and the other controls the scrollable text field.

Figure 1.8: The interactive elements in this Flash interface are generated using components.

The navigation is controlled by a component called Springy Thingy. It looks like a standard, vertically oriented navigation system. You can roll over the buttons, and each button changes color as you do. However, you can also drag a button, and the other buttons will move along with it in a fun bouncy or springy motion (see Figure 1.9). The scrollable text component controls the functionality that lets you click the up or down arrows or drag the bar to view all of the text in the field.

Figure 1.9: Dragging the navigation bar around produces a bouncy or springy animation effect.

Figure 1.10 shows the interface within Flash MX. The gray box with the Springy Thingy label is the component used to create the navigation system. Since components are created from standard movie clips, they can look like just about anything when you are working with them in Flash. In Figure 1.10, the gray box with the component icon to the left is Eyeland Studio's standardized look for our components. This visual treatment helps to set it apart from other resources on the stage, but it is not representative of what all components will look like.

Figure 1.10: One of the components is easier to pick out than the other.
The scroll bar to the right of the "You Need a Vacation!" field is also a component. The scroll bar component is more indicative of the kind of component you are likely to find on the Web.

Components that you find on the Internet are commonly hard to identify as components when you are working with them in Flash. There is no standard for how components look when they are on the stage. See later in this chapter for some sites that distribute components.

Figure 1.11 shows the component user interface for the Springy Thingy component. The controls in the interface let you adjust various parameters in the component. In the case of Springy Thingy, different parameters allow you to specify the text for each button, the colors for the text, the colors for the various states of the buttons, the shape of the buttons, the size of the buttons, and much more.

Figure 1.11: The user interface for Springy Thingy provides a quick and easy way for customizing the characteristics of the component.

While the user interface might look a little intimidating at first, it's actually not that much harder to use than any dialog box or panel in Flash. If you are familiar with how to adjust font characteristics for text in Flash, then you should have very little trouble using a component UI like this. You will get plenty of hands-on experience using component UIs similar to this one, in later chapters.

The advantage of using components becomes clear when you consider what it would take to generate the same functionality by coding from scratch. Scott Balay from Eyeland Studio coded the Springy Thingy component. Scott has over 10 years professional programming experience and has been programming with ActionScript ever since it was introduced in Flash 5.

Programming the essential elements of the Springy Thingy navigation system took Scott about six hours, including testing and troubleshooting. This does not include time spent creating the actual Springy Thingy component or the component user interface. Scott readily admits that some programmers might be able to work faster; however, it would still take most Flash programmers several hours to program the overall navigation system from scratch.

When I created this Flash interface template for Eyeland Studio's Flash Foundry, it took me only 15 or 20 minutes to set the parameters for the Springy Thingy component so that it looked good in context with the rest of the artwork in the template. Much of this time was actually taken up by making an adjustment, testing to view the results, and then going back to make a few more adjustments.

Obviously, 15 to 20 minutes is better than six hours. Relatively inexperienced programmers might take days to program similar functionality. However, by using a component to generate the functionality, I was able to use those hours and days for other things (such as sleep, having a life, and other previously inconceivable things).

Let's look at one more example. Figure 1.12 shows a concentration game for the Musco Family Olive Company (www.olives.com). The Olives site design and the artwork for the game was done by the Tesser agency (www.tesser.com). The game is a classic concentration or match game with four levels. Each level increases in difficulty and in the points earned for completion.

Figure 1.12: This concentration game for the Musco Family Olive Company was created using components. Brandon Williams did the programming for this game. Brandon is also an advanced programmer who has been published in several books. It took Brandon approximately 12 hours to program this game. Once again, it took me about 15 to 20 minutes to change the parameters for this game using the component.

This component, however, is a little different than the Springy Thingy example. The concentration game component works with art resources extracted from the library. The art that works with the component is used for the game pieces as well as the pictures that are revealed when the player makes matches. Although setting up these game pieces to work with the component takes some time, it is not nearly as cumbersome as it would be without the component.

Figure 1.13 shows the concentration game component in Flash with its user interface open. Notice that there are no game pieces on the stage for the game. The artwork for the game is not on the stage. This is because the component actually extracts the art resources from the library. Since there are several levels to the game, this helped keep the game nice, neat, and tidy during production. In other words, using the component, in this case, made this game easier to generate because I didn't have to go to the trouble of programming it and because it was more manageable.

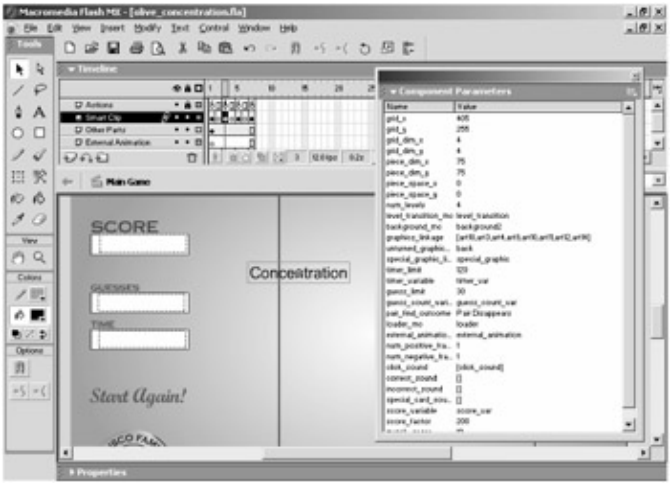

Figure 1.13: There is an instance of the concentration for each level of the game. Each instance has the parameters set differently.

Team LiB
Team LiB

Where Can I Find Components?

At the time of this writing, there aren't too many sites devoted to components, yet more and more sites are popping up on the Internet. Some developers create components and distribute them for free to help promote themselves. An increasing number of sites are beginning to sell components. Components can range from doing very small tasks such as powering a radio button to generating highly elaborate navigational systems of customizable games. The following is a list of places where you can find components:

Macromedia Macromedia's site hosts Flash Exchange. The link is dynamic. The easiest way to find Flash Exchange is to go the Products section at macromedia.com. Select Flash and then click on the Flash Exchange link. Flash Exchange features free components and a rating system so that you can get some idea of what other people think of the component. Since they are free, most of the components in Flash Exchange are modest in scope. You won't find previews for the components, but you can join a public forum for Flash Exchange where you can benefit from the knowledge of other component developers and component users.

- www.macromedia.com

Flash Foundry, Eyeland Studio Flash Foundry is a membership-based product. Members purchase access to components and other Flash resources. The components in Flash Foundry utilize a highly developed custom UI. You can preview the components and enter a public forum for support and information exchange. Flash Foundry also features very detailed interactive Flash tutorials that cover topics ranging from specific components to general tutorials that apply to any number of components. Flash Foundry is updated frequently with more components. Most of the components featured in this book are modified versions of components found at flashfoundry.com.

- www.flashfoundry.com

FlashComponents.com, Art Today FlashComponents.com is based on the same components and flash resources as the previously mentioned site. However, FlashComponents.com is operated by Art Today, the company behind clipart.com, photos.com, and rebelartists.com. FlashComponents.com has most of the same benefits mentioned above; however, from time to time members can also benefit from special offers for Art Today's other membership products.

- www.flashcomponents.com

Flashkit.com Flashkit offers several free components and smart clips in their free movies section. Flashkit does only rudimentary screening of the resources placed on their site, which means the quality of the resources can be somewhat unimpressive. However, free is free, and you usually get what you pay for.

- www.flashkit.com

Flashcomponents.net Flashcomponents.net is one of the first sites devoted to components on the Internet. The components are free, although some might have usage restrictions. Flashcomponents.com has previews, and the quality of the components are above average—particularly for a free site.

- www.flashcomponents.net

Games in a Flash Games in a Flash offers a growing collection of component-based Flash game engines for licensing. The games are customizable via the components, and the graphics and audio in the games can be edited to customize the look of the games. The game engines come with interactive Flash tutorials that demonstrate how to customize the games.

- www.gamesinsaflash.com

Methinks Methinks offers a small collection of components for Flash that generate special effects such as rotation, zoom, and fading.

- www.methinks.com

Not listed above are personal sites that offer free components (although many of those same components are available at flashcomponents.net or Flash Exchange). There is no shortage of free components on the Internet, but the trade-off is that they usually do not utilize user-friendly, custom user interfaces, and there is usually no guarantee of any kind of support. Components that you pay for are typically held to a higher standard, and they almost always come with at least some level, if not multiple levels, of support (such as tutorials, built-in help information, FAQs, forums, e-mail support, and even phone support).

Team LiB
Team LiB

Conclusion

Components free you from concern with how you are going to implement the advanced functionality for a project. Furthermore, using components allows you to keep your projects from getting out of hand. These in turn free you up to do things like focus more on generating effective designs, creating compelling content, and maybe even meeting your deadlines with time to spare (cue laugh track).

While we will be looking at how to *create* components later, in Chapters 10 and 11, the majority of this book is about how to *use* components. We go step-by-step through the process of using many different types of components. Whether you are a beginner or an advanced Flash user, learning how to grab the power of components can be very worthwhile. If there is a component available that performs the functionality you need, then-as the saying goes- there's usually no point in reinventing the wheel.

In the next chapter, we will look at the basics of using components. While components have some similarities to plug-ins or macros, components have their own set of characteristics-some of them are even a bit on the quirky side. However, a little hands-on experience with components will help you to see their incredible potential.

Team LiB
Team LiB

Chapter 2: Using Components 101

Overview

Macromedia has constructed Flash MX with a few little features that can easily trip up the uninitiated. In addition, many of the same aspects that make components versatile and powerful can also make them slightly confusing to work. If you know what to look for and what to watch out for, components can be extremely valuable resources.

The best way of overcoming these minor hurdles is to simply start working with components. In this chapter, you will work with a relatively simple component that utilizes the default component user interface for the Component Parameters panel, exploring some of the advantages and disadvantages (mostly disadvantages) of the default UI. I'll also explain the parameters you'll encounter and their types.

Most of this book's components use custom user interfaces. However, components available for free frequently utilize the default UI. In addition to learning the basics of components, working with the default UI will help you appreciate the value of good custom UIs.

- **The default component UI**
- **Understanding parameter types and value types**
- **Experimenting with values**

Team LiB
Team LiB

Parameters and the Default Component User Interface

Fundamentally, a component allows you to manipulate variables within ActionScript without having to delve into the actual ActionScript. For instance, let's say you had some ActionScript that utilizes Flash MX's ability to draw a rectangle with code. Even something as simple as a rectangle has several potential variables. How high is the rectangle? How wide? Where is the rectangle on the stage? What color is it? What is its alpha?

All of these variables can be "hard-coded" within the ActionScript, meaning that the variables are part of the code. The problem with this is that if you want to change these variables, you must go into the code. Finding and editing the code required to generate a rectangle would not be particularly hard. However, components that perform more complex actions like creating a collapsible menu or a game contain code that can be hundreds of lines long.

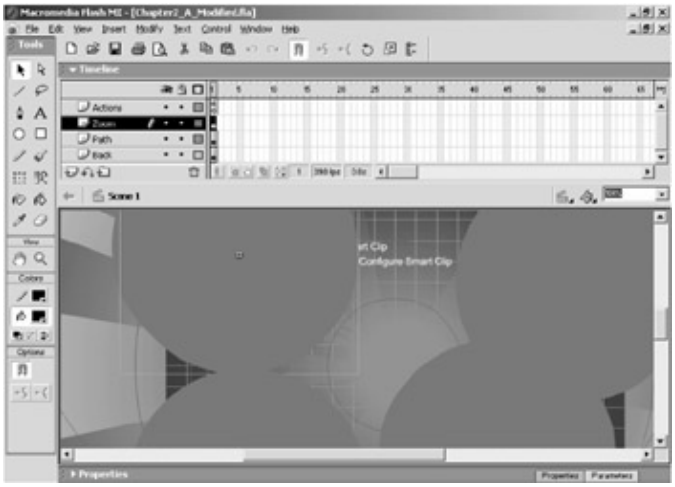

Figure 2.1: The Chapter2_A_Modified.fla sample file is dotted with multiple instances of two separate components.

Furthermore, the code might not all be in one place. For complex resources, finding and editing the variables can be difficult and time-consuming even for experienced programmers-even for the person who wrote the code, particularly if they need to change the variables a long time after they originally wrote it. Components provide a solution to this problem and, in the process, make the code easier to customize, for the programmer and nonprogrammer alike.

In components, each variable is called a *parameter*. Components make it easier to customize or change the parameters by placing them in a more accessible location: the Component Parameters panel. As you will see in the following example, the Component Parameters panel can take on a wide range of different appearances.

On the CD Open Chapter2_A_Start.fla in the Chapter 2 folder on this book's CD. Save the file to your local hard drive as Chapter2_A_Modified.fla.

In this Flash file, the stage is peppered with components. Actually, there are only two components, but there are multiple instances of each of the two components. An *instance* refers to a particular copy of a symbol in Flash. For example, if you create a movie clip and then drag three copies of that movie clip onto the stage in Flash, you have three instances of that movie clip.

You can manipulate components in much the same way you can manipulate any other symbol in Flash. You can generate multiple instances of a component, and you can duplicate and/or copy a component. If you use multiple instances of a component, you gain the same file-size-saving advantages that other symbols provide.

Go ahead and test the movie to observe what it does (see Figure 2.2). What you will see is a rich animation. If you look carefully, you'll notice two types of animations, both of which utilize simple circles. A series of circles marches end to end, and lines of circles march up, down, left, and right in all different sizes and varied opacities. Each line of marching circles is controlled by a component. Notice, also, the series of circles that zoom in and out of view. Some zoom in, some zoom out. Each zooming circle is also controlled by a component. Close the test movie.

Figure 2.2: Multiple copies of two components are used in this example to generate a rich, textured, animated background.

Team LiB
Team LiB

Exploring the Component Parameters Panel

Go to the Zoom layer, select one of the large red circles, and open the Component Parameters panel. You can open the panel in several different ways, but each requires that you first select the component on the stage. Once you have selected the component, you can choose Windows &"internaljump" href="#ch02fig03">Figure 2.3). You can also right-click or Control+click a component and then select Component Parameters from the Panels submenu (see Figure 2.4). Probably the easiest method for opening the Component Parameters panel, however, is to use the shortcut key: Option/Alt+F7.