

## Preface

This book teaches both ActionScript's fundamentals and its advanced usage. Over the next 675 meaty pages, we'll be exploring every detail of the ActionScript language—from the basics of variables and movie clip control to advanced topics such as objects and classes, server communication, and XML. By the end, we'll have covered everything there is to know about Flash programming.

This book is not just for programmers. The text moves pretty quickly, but a prior knowledge of programming is not required to read it. All you need is experience with the non-ActionScript aspects of Flash and an eagerness to learn. Of course, if you are already a programmer, so much the better; you'll be applying your code-junkie skills to ActionScript in no time.

This book fully documents the material that may be undocumented or underdocumented by Macromedia or by other third-party books. Flash is notorious for word-of-mouth techniques and esoteric features. How are layers, movie clips, and loaded movies stacked in the Player? (See Chapter 13, *Movie Clips*.) What governs the execution order of code on any given frame? (See Chapter 13.) Do event handlers have a local scope? (See Chapter 10, *Events and Event Handlers*.) Why does the number 90 sometimes show up as 89.9999999997? (See Chapter 4, *Primitive Datatypes*.) It has been my particular goal to chart these unknown waters. Of course, I also cover the basic programming techniques required in any language, such as how to make a segment of code execute repeatedly. (See Chapter 8, *Loop Statements*.)

This book is designed to be kept on your desk, not to wallow away its life on your shelf. Part III, *Language Reference*, exhaustively covers every object, class, property, method, and event handler in ActionScript. You'll use it regularly to learn new things and remind yourself of the things you always forget.

Above all, this book is a Definitive Guide. It's the product of years of research, thousands of emails to Macromedia employees, and feedback from users of all levels. I hope that it is self-evident that I've suffused the book with both my intense passion for the subject and the painfully won real-world experience from which you can benefit immediately. It covers ActionScript with exhaustive authority and—thanks to a technical review by Gary Grossman, the creator of ActionScript—with unparalleled accuracy.

## *What Can ActionScript Do?*

Frankly, there's no practical limit to what a full-fledged language like Flash 5 ActionScript can achieve. Let's take a look at some of ActionScript's specific capabilities to offer a taste of the topics we'll cover throughout the book. Begin thinking about how you can combine these techniques to accomplish your particular goals.

### *Timeline Control*

Flash movies are composed of frames residing in a linear sequence called the *timeline*. Using ActionScript, we can control the playback of a movie's timeline, play segments of a movie, display a particular frame, halt a movie's playback, loop animations, and synchronize animated content.

### *Interactivity*

Flash movies can accept and respond to user input. Using ActionScript, we can create interactive elements such as:

- Buttons that react to mousedownclicks (e.g., a classic navigation button)
- Content that animates based on mouse movements (e.g., a mouse trailer)
- Objects that can be moved via the mouse or keyboard (e.g., a car in a driving game)
- Text fields that allow users to supply input to a movie (e.g., a fill-in form)

### *Visual and Audio Content Control*

ActionScript can be used to examine or modify the properties of the audio and visual content in a movie. We may, for example, change an object's color and location, reduce a sound's volume, or set the font face of a text block. We may also modify these properties repeatedly over time to produce unique behaviors such as physics-based motion and collision detection.

## *Programmatic Content Generation*

Using ActionScript, we can generate visual and audio content directly from a movie's Library or by duplicating existing content on the Stage. Programmatically generated content may serve as a strictly static element, such as a random visual pattern, or as an interactive element, such as an enemy spaceship in a video game or an option in a pull-down menu.

## *Server Communication*

ActionScript provides a wide variety of tools for sending information to and receiving information from a server. The following applications all involve server communication:

- Link to a web page
- Guest book
- Chat application
- Multiplayer networked game
- E-commerce transaction
- Personalized site involving user registration and login

These examples, of course, offer only a limited account of potential ActionScript applications. The goal of this book is to give you the fundamental skills to explore the myriad other possibilities on your own. This is not a recipe book—it's a lesson in cooking code from scratch. What's on the menu is up to you.

## *The Code Depot*

We'll encounter dozens of code samples over the upcoming chapters. To obtain relevant source files and many other tutorial files not included in the book, visit the online Code Depot, posted at:

*<http://www.moock.org/asdg>*

The Code Depot is an evolving resource containing real-world ActionScript applications and code bases. Here's a selected list of samples you'll find in the Code Depot (you can download them individually or as a single *.zip* file):

- A multiple-choice quiz
- An XML-based chat application
- A guest book application
- A custom mouse pointer and button

- An asteroids game code base
- Programmatic motion effects
- Demos of HTML text fields
- Preloaders
- String manipulation
- Interface widgets, such as slider bars and text scrollers
- Mouse trailers and other visual effects
- Volume and sound control

Additionally, any book news, updates, tech notes, and errata will be posted at the preceding URL as well as on the book's web site.

## Showcase

Practically every Flash site in existence has at least a little ActionScript in it. But some sites have, shall we say, more than a little. Table P-1 presents a series of destinations that should provide inspiration for your own work. See also the sites listed in Appendix A, *Resources*.

Table P-1. ActionScript Showcase

Topic	URL
Experiments in design, interactivity, and scripting	<a href="http://www.yugop.com">http://www.yugop.com</a> <a href="http://www.prystation.com">http://www.prystation.com</a> * <a href="http://www.presstube.com">http://www.presstube.com</a> <a href="http://www.pitaru.com">http://www.pitaru.com</a> <a href="http://www.flight404.com">http://www.flight404.com</a> <a href="http://www.bzort-12.com">http://www.bzort-12.com</a> <a href="http://kaluzbny.nm.ru/3D.html">http://kaluzbny.nm.ru/3D.html</a> * <a href="http://www.protocol7.com">http://www.protocol7.com</a> * <a href="http://www.uncontrol.com">http://www.uncontrol.com</a> * <a href="http://www.digitalnotions.com/dev/flash5">http://www.digitalnotions.com/dev/flash5</a> * <a href="http://flash.onego.ru">http://flash.onego.ru</a> * <a href="http://www.figleaf.com/development/flash5">http://www.figleaf.com/development/flash5</a> *
Games	<a href="http://www.gigablast.com">http://www.gigablast.com</a> <a href="http://www.sadisticboxing.com">http://www.sadisticboxing.com</a> <a href="http://www.flashkit.com/arcade">http://www.flashkit.com/arcade</a> * <a href="http://www.huibui.de">http://www.huibui.de</a>
Interface and dynamic content	<a href="http://www.mnh.si.edu/africanvoices">http://www.mnh.si.edu/africanvoices</a> <a href="http://www.curiousmedia.com">http://www.curiousmedia.com</a>

\* Downloadable *.fla* files provided. Otherwise only *.swf* files available.

## Typographical Conventions

In order to indicate the various syntactic components of ActionScript, this book uses the following conventions:

- **Constant width** for code samples, clip instance names, frame labels, property names, and variable names.
- *Italic* for function names, method names, class names, layer names, filenames, and file suffixes such as *.swf*.
- **Constant width bold** for code that you must type when following a step-by-step procedure.
- *Constant width italic* for code that you must replace with an appropriate value (e.g., *your name here*) or for variable and property names referenced in a code comment.
- Method and function names are followed by parentheses.

Pay special attention to notes set apart from the text with the following icons:



This is a tip. It contains useful supplementary information about the topic at hand.



This is a warning. It helps you solve and avoid annoying problems.

## We'd Like to Hear from You

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

O'Reilly & Associates, Inc.  
101 Morris Street  
Sebastopol, CA 95472  
(800) 998-9938 (in the U.S. or Canada)  
(707) 829-0515 (international/local)  
(707) 829-0104 (fax)

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

*<http://www.oreilly.com/catalog/actscript>*

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, software, Resource Centers, and the O'Reilly Network, see our web site at:

*<http://www.oreilly.com>*

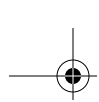
## Acknowledgments

I am honored to know, privileged to work with, and deeply indebted to the following people:

- The tremendously talented Flash team at Macromedia, who through constant innovation have shaped a medium, and who recognize the people behind the terminals that constitute the Web. Macromedia's combination of professionalism, exploration, and personal passion is rare within the construct of a corporation.
- The consummate professionals at O'Reilly: Tim O'Reilly, Troy Mott, Mike Sierra, Rob Romano, Edie Freedman, and the many copyeditors, indexers, proofreaders, and sales and marketing folks who helped bring this book to the shelves.
- Derek Clayton, my personal programming mentor and friend. In addition to providing almost-daily code advice, Derek contributed the Perl code in Chapter 17, *Flash Forms*, and schooled me in Quake far too many times. He also wrote an *XMLSocket* server in Java and a generic flat file database system in Perl, both available from the online Code Depot.
- Wendy Schaffer, who, in addition to proofreading the first draft of the manuscript, bore the task of supporting me with love and life during the all-consuming task of writing this book.
- Bruce Epstein, who served as the developmental editor, helping to refine nearly every sentence in the manuscript, often contributing content to round out topics. Bruce's insightful editorial work and guidance were invaluable.
- Gary Grossman, the creator of ActionScript at Macromedia, who somehow always found time to answer questions, explain subtleties, and even entertain debates during ActionScript's development. Gary acted as lead technical editor

for the book, helping to clarify important concepts and details. Much of the accuracy of this text is the direct result of his participation.

- Slavik Lozben, Macromedia Flash Engineer, whom I cannot thank enough for creating movie clip events and *swapDepths*! Without Slavik's intelligence and willingness to engage in discourse, I'd still be writing the chapter on events and event handlers. Slavik also contributed greatly as a technical editor.
- Erica Norton, Macromedia ActionScript QA Engineer, who answered and researched question after question with alacrity. In addition to entertaining regular discussions, Erica took time out of her demanding schedule to be a technical editor.
- Jeremy Clark, Macromedia's Flash Product Manager, who enthusiastically supported the book with ideas, advice, friendship, and answers to my endless stream of questions. Eric Wittman, Macromedia's Director of Flash Product Management, whose sagacious direction has shaped Flash for years. Janice Pearce, of Macromedia's Flash QA team, who clarified various Flash production issues and graciously supplied early builds of Flash 5. Matt Wobensmith, Macromedia's Flash Community Manager, and a regular and valuable source of knowledge. Troy Evans, Macromedia's Flash Player Product Manager, who guides and champions the Flash Player. Bentley Wolfe, from Macromedia's technical support team, who seems to never leave his keyboard.
- Richard Koman of O'Reilly & Associates, who provided editorial guidance during the proposal and early draft stages.
- David Fugate, my literary agent from Waterside Productions, whose diligence and confidence made business affairs practically invisible.
- D. Joe Duong, who knows too much for his age, in a kind of defiance of the amount of time he spends teaching others. I am lucky enough to be one of those others. Mike Linkovich, the code philosopher, who is as inspiring as he is edifying. James Porter and Andrew Murphy for reading and testing, and reading and testing, and reading and testing. And, of course, Graham Barton, the deserter.
- Doug Keeley, Terry Maguire, and Jon Nicholls, who created ICE, a company where occupation and passion are allowed to coexist.
- Professor Paul Beam, who saw the connection between literature, communication, and computers in the unlikely setting of an English graduate program at the University of Waterloo.
- Professor Jack Gray, for his wisdom and friendship.
- Andrew Harris, David Luxton, Michael Kavanagh, Stephen Burke, Cheryl Gula, Christine Nishino, Stephen Mumby, Karin Trgovac, and Judith Zissman, whom I hold in esteem for their art, ideas, and friendship.



- The Flash community, from whom I draw inspiration and understanding, including James Patterson, Yugo Nakamura, Naoki Mitsuse, Joshua Davis, James Baker, Marcell Mars, Phillip Torrone, Robert Reinhardt, Mark Fennell, Branden Hall, Josh Ulm, Darrel Plant, Todd Purgason, John Nack, Jason Krogh, Hillman Curtis, Glenn Thomas, and whomever I've inevitably omitted.
- The Moocks (Margaret, Michael, Jane, and Biz) for teaching me to think, dream, explore, and love.
- The Schaffers, for years of family and friendship.

And lastly I'd like to thank you the reader for taking the time to read this book. I hope it helps to make my passion your own.

—Colin Moock  
Toronto, Canada  
April 2001

