# The Essential Guide to Flex 2 with ActionScript 3.0

Charles E. Brown



an Apress<sup>®</sup> company

### The Essential Guide to Flex 2 with ActionScript 3.0

Copyright © 2007 by Charles E. Brown

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-733-0

ISBN-10 (pbk): 1-59059-733-8

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit www.apress.com.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

#### Credits

Lead Editor Assistant Production Director Chris Mills Kari Brooks-Copony

Technical Reviewer Sas Jacobs

Laura Cheu Compositor

Diana Van Winkle

Senior Production Editor

#### **Editorial Board**

Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Matt Wade

**Artist** April Milne

Proofreader Liz Welch

Senior Project Manager Beth Christmas

**Indexer** Toma Mulligan

Copy Edit Manager Nicole Flores

Interior and Cover Designer Kurt Krames

Copy Editor Ami Knox

Manufacturing Director Tom Debolski

# CONTENTS AT A GLANCE

About the Author	<b>xi</b>
About the Technical Reviewer	. xiii
Acknowledgments	. <b>XV</b>
Introduction	xvii
Chapter 1: Introducing RIAs and Installing Flex Builder 2	1
Chapter 2: Introducing Flex and Flex Builder 2	. 23
Chapter 3: Introducing ActionScript 3.0	. 47
Chapter 4: Containers and Components	. <b>89</b>
Chapter 5: Navigation Containers	135
Chapter 6: Flex and XML	151
Chapter 7: States	209
Chapter 8: Case Study: Part 1	243
Chapter 9: The Repeater Component	295
Chapter 10: Case Study: Part 2	313
Chapter 11: Drag and Drop	369
Chapter 12: Flex and ColdFusion	387
Chapter 13: Charting	431
Chapter 14: Printing	453
Chapter 15: Unfinished Symphony	467
Appendix: Installing ColdFusion MX 7	471
Index.	483

# CONTENTS

About the Author	<b>X</b> i
About the Technical Reviewer	xiii
Acknowledgments	<b>XV</b>
Introduction	. <b>xvi</b> i
Chapter 1: Introducing RIAs and Installing Flex Builder 2	1
Understanding the ActionScript 3.0/Flex 2 environment	2 4 6
Installing Flex Data Services Express	10 18 22
Chapter 2: Introducing Flex and Flex Builder 2	23
Web design and Flex	24 24 25
ActionScript 3.0	26
Flex and the Flex Builder environment       Creating your first Flex project	27
Working in Source view	33
Why Flex?	

#### CONTENTS

Chapter 3: Introducing ActionScript 3.0	47
Creating an ActionScript 3.0 project	48
Object-oriented programming concepts	
Understanding what a class file is	50
Inheritance	52
	52
ActionScript 2.0 Language Deference	JZ
ActionScript 5.0 Language Reference	
Object-oriented system design	5/
Your first class files	5/
	5/
Giving the class file functionality	60
Comments	60
Using the trace() method	61
Escape sequence	64
Combining MXML and ActionScript 3.0	65
Layout containers	65
Adding controls	68
Binding controls	69
Adding ActionScript code	72
ActionScript constructs	
Valiables	
	/5
	/5
Functions	/6
Passing parameters	/8
Handling events	81
Reusability	87
Summary	88
Chapter 4: Containers and Components	<b>89</b>
Creating a project	90
Main application file	95
Building within the main container	96
Debugging in Elex Builder 2	96
Using containers	103
	113
	113
Dinding with ActionScript	110
	119
	122
	123
Creating the component	124
Calling components	126
Passing parameters	128
Receiving parameters from a component	130
Cleaning up a project	132
Summary	134

Chapter 5: Navigation Containers	5
Working with navigation containers	6
ViewStack navigation container	12
The LinkBar and TabBar controls	13
Accordion navigation container	16
TabNavigator navigation container    14	17
Adding another container	8
Summary	0
Chapter 6: Flex and XML 15	1
Data source	52
Events	;3
The event object	57
XML: A very brief introduction	51
Using XML in Flex	53
The Model tag	56
Working with an external XML file	58
Using the HTTPService tag	59
Displaying data	'1
XML and ActionScript	2'
When good code goes bad	7
Flash Player security	31
Introducing E4X	33
Using the DataGrid control	38
Modifying DataGrid columns	39
Changing date format	)0
Editing and rendering data	)2
Applying the DataGrid container	)0
Summary	17
Chapter 7: States	9
Understanding states	0
Changing the state	6
States and code	22
Rollovers and states	24
Transitions	31
Building the container	32
Building the states	34
Creating transitions	37
Summary	1

Chapter 8: Case Study: Part 1	. 243
The case study: a book publisher's website	244
Creating the project environment	245
Starting to build the structure	252
Creating the components	254
BookHome component	254
Comments component	262
BookCovers component	267
Assembling the components	273
BookCart component	280
Flex and CSS	288
CSS: The basics	289
Working with CSS in Flex	290
Summary	293
Chapter 9: The Repeater Component	. 295
Understanding the Repeater component	296
Passing data in a Repeater component	299
Using XML data	303
Changing component properties	310
Summary	311
Chapter 10: Case Study: Part 2	. 313
Chapter 10: Case Study: Part 2	. <b>313</b> 314
Chapter 10: Case Study: Part 2	. <b>313</b> 314 316
Chapter 10: Case Study: Part 2	. <b>313</b> 314 316 316
Chapter 10: Case Study: Part 2	. <b>313</b> 314 316 316 324
Chapter 10: Case Study: Part 2 Setup Coding the application Connecting the data source Testing your code Handling the book cover images	. <b>313</b> 314 316 316 324 326
Chapter 10: Case Study: Part 2 Setup Coding the application Connecting the data source Testing your code Handling the book cover images Changing states	. <b>313</b> 314 316 316 324 326 335
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart	. <b>313</b> 314 316 316 324 326 335 346
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event	. <b>313</b> 314 316 316 324 326 335 346 348
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class	<b>313</b> . 314 . 316 . 316 . 324 . 326 . 325 . 346 . 348 . 353
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class         Putting the pieces all together	<b>313</b> . 314 . 316 . 316 . 324 . 326 . 325 . 346 . 348 . 353 . 359
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Testing your code         Handling the book cover images         Changing states         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component	<b>313 314 316 316 324 326 335 346 348 353 348 353 359 365</b>
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component         Summary	<b>313 314 316 316 324 326 335 346 348 353 359 365 368</b>
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component	<b>313 314 316 316 324 326 335 346 348 353 359 365 368</b>
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component         Summary	<b>313 314 316 316 324 326 335 346 348 353 359 365 368 369</b>
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component         Summary         Understand the drag-and-drop concepts	. 313 . 314 . 316 . 316 . 324 . 326 . 325 . 346 . 348 . 353 . 359 . 365 . 368 . 369 . 370
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component         Summary         Understand the drag-and-drop concepts         Dragging to a DataGrid	. 313 . 314 . 316 . 316 . 324 . 326 . 325 . 346 . 348 . 353 . 359 . 365 . 368 . 369 . 370 . 371
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component         Summary         Understand the drag-and-drop concepts         Dragging to a DataGrid         Dragging to a List control	. 313 . 314 . 316 . 324 . 326 . 325 . 346 . 348 . 353 . 348 . 353 . 359 . 365 . 368 . 369 . 370 . 371 . 377
Chapter 10: Case Study: Part 2         Setup         Coding the application         Connecting the data source         Testing your code         Handling the book cover images         Changing states         Finishing the shopping cart         Creating a class for a custom event         Customizing your class         Putting the pieces all together         The TileList component         Summary         Understand the drag-and-drop concepts         Dragging to a DataGrid         Dragging to a List control         Nonenhanced drag and drop	. 313 . 314 . 316 . 324 . 326 . 325 . 346 . 348 . 353 . 348 . 353 . 359 . 365 . 368 . 368 . 370 . 371 . 377 . 381

#### CONTENTS

Chapter 12: Flex and ColdFusion
Introducing ColdFusion388Installing a data source389Connecting Flex and ColdFusion393Dissecting and modifying your application407Variations on a theme414Configuring for ColdFusion424Summary430
Chapter 13: Charting
Understand the Flex charting components432Using the PieChart component434Using the ColumnChart component441Animating the chart447Using the BarChart component450Summary452
Chapter 14: Printing
The printing process454Creating a separate Print container457Printing and components462Summary466
Chapter 15: Unfinished Symphony
Conclusion
Appendix: Installing ColdFusion MX 7
Index

### **ABOUT THE AUTHOR**



**Charles E. Brown** is one of the most noted authors and teachers in the computer industry today. His first two books, *Beginning Dreamweaver MX* and *Fireworks MX Zero to Hero*, have received critical acclaim and are consistent bestsellers. In early 2004, Charles coauthored a book on VBA for Microsoft Access—*VBA Access Programming*.

In addition to his busy writing schedule, he conducts frequent seminars as an Adobe Certified Trainer. His topics include Flex, Flash, Dreamweaver, and ActionScript programming.

He is also frequently called in as a consultant for major websites involving Adobe technologies.

Charles is also a noted classical organist, pianist, and guitarist, and studied with such notables as Vladimir Horowitz, Virgil Fox, and Igor Stravinsky. It was because of his association with Stravinsky that he got to meet, and develop a friendship with, famed artist Pablo Picasso. Charles can be contacted through his website at www.charlesebrown.net.

### ABOUT THE TECHNICAL REVIEWER



**Sas Jacobs** is a web developer who set up her own business, Anything Is Possible, in 1994, working in the areas of web development, IT training, and technical writing. The business works with large and small clients building web applications with .NET, Flash, XML, and databases.

Sas has spoken at such conferences as Flashforward, webDU (previously known as MXDU), and FlashKit on topics related to XML and dynamic content in Flash. In her spare time, Sas is passionate about traveling, photography, running, and enjoying life.

### ACKNOWLEDGMENTS

This has been a long journey.

I heard the first murmurings of Flex 2 early in 2005. I never felt that Flex 1, due to the limitations and quirkiness of ActionScript 2.0, was a very viable product. Based on sales, many felt as I did.

I was not surprised that Flex 2 would mean a completely restructured ActionScript. It was also apparent to me why the Adobe/Macromedia merger had to happen.

In the fall of 2005, the full beta for Flex 2 was unveiled at the MAX convention in Anaheim, California. This also marked the beginning of the Adobe policy of public betas.

About that time, Chris Mills, my editor at friends of ED, approached me about doing a book on the subject. What I thought would be the easiest job in the world turned out to be the most difficult I have ever encountered. Between the beta's first release and the final product release in the summer of 2006, there were several major overhauls. This required that we throw out huge sections of the book and start over. I can honestly say we wrote this book three times before we arrived at what you hold in your hands right now.

It also meant taking nearly everything I knew about ActionScript and throwing it out. In other words, I had to relearn this knowledge myself.

Along the way, I wanted a book that could adapt with the inevitable changes that will come. As a result, the site www.charlesebrown.net came into being to continue this book long past publication.

I couldn't have done it alone, and some thanks are in order:

First of all, I want to thank my editor, Chris Mills, for his wisdom and guidance, as well as many mornings on Messenger listening to me gripe about what a tough book this was to write.

Every time I thought I wrote the perfect chapter, Sas Jacobs, my technical editor, brought me back to reality. This book would not have been possible without her guidance and wisdom.

#### ACKNOWLEDGMENTS

I have to thank my project manager, Beth Christmas, twice: First for persistently nudging me back on schedule every time I slacked off. Second, for a wonderful dinner in Philadelphia (with some friends) on a lonely night when I was conducting a training seminar there. Beth, I still owe you dinner at the City Tavern.

Finally, I want to thank all of my many friends and supporters of this project (including some students at my training classes) for their invaluable suggestions and insights.

I hope this book brings you, the reader, the knowledge you need to be a successful Flex programmar.

### INTRODUCTION

Welcome to the future!

No, I am not making some metaphysical statement here. Instead, it is a statement about how many of us will build websites in the future. There is no doubt that Flash is playing an increasing role in web design, and Flex is the next evolutionary step for Flash.

In many ways, I struggled with this book. Was it to be a book about rich Internet websites? Was it to be a book about the MXML language that Flex uses? Was it to be a book on the next generation of ActionScript, ActionScript 3.0? Instead, I tried to make it a book that takes a very broad view of all three disciplines and how they fit together.

Let me start off by saying that many of the explanations are uniquely mine. After years of doing technical training, where I have only a few days to cover large topics, I have learned to substitute shorter explanations that clarify a concept in place of larger more technical (and often confusing) explanations. In other words, I often like to get right to the point without taking circuitous routes.

Please keep a few things in mind when reading this book. First, you will find that the techniques I show you are techniques that reflect my style of programming and design. Certainly there are many alternative ways of arriving at the same point. It is impossible for any one book to cover all possible variations, especially with topics as large as I cover here. If you find a different way of doing something, by all means use it if it works for you.

Second, I very purposely kept my examples simple in order to illustrate a point. I do not want you, the reader, to get caught up in just following recipe-like instructions that do little more than test your ability to follow instructions. While I have a case study in this book, it is far from complete. I have given details of a website at the end of this book where I am going to invite you, the reader, to submit various solutions and to join in various discussions. I hope you will take advantage of this.

Third, I am assuming that you already have at least a cursory knowledge of object-oriented programming concepts. While I do review these concepts in earlier chapters, it is only a very basic introduction. OOP is a very large subject in which large volumes have been written.

OK, enough of the warnings and disclaimers.

What I hope this book does is give you enough of a taste of Flex and the ActionScript 3.0 environment that you will be able to solve the unique problems your own situations will require. I spend a great deal of time discussing how to find help by using the ActionScript 3.0 Language Reference, for example.

If you have ever used ColdFusion, you probably know that it uses a simple language (CFML) to write a more complex language in the background (Java). Essentially, Flex does exactly the same thing: it uses MXML, the Flex equivalent of CFML, to write more complex ActionScript 3.0 in the background. In this book, I try to show you how to accomplish the same tasks in MXML and ActionScript 3.0.

I hope you walk away from this book with the same sense of excitement that I have about Flex 2. I really encourage you to experiment and study further. Look upon this book as the beginning, not the end.

On to the future.

#### Layout conventions

To keep this book as clear and easy to follow as possible, the following text conventions are used throughout:

Important words or concepts are normally highlighted on the first appearance in **bold type**.

Code is presented in fixed-width font.

New or changed code is normally presented in **bold fixed-width font**.

Menu commands are written in the form Menu ➤ Submenu. ➤ Submenu.

Where I want to draw your attention to something, I've highlighted it like this:

Ahem, don't say I didn't warn you.

Sometimes code won't fit on a single line in a book. Where this happens, I use an arrow like this:  $\Rightarrow$ .

This is a very, very long section of code that should be written all on the same  $\blacktriangleright$  line without a break.

# 1 INTRODUCING RIAS AND INSTALLING FLEX BUILDER 2

friendsof	Click date, child	butto check ren.	ns bel k-out d	ow to late, re	select ooms,	che adul	🖳 Adobe Flex Data Servi
	Sun	Octob Mon	er 200 Tue	6 - No Wed	vembe Thu	er 20 Fri	
an Apress- company	1	2	3	4	5	6	Welcome
hama hasha daumlaa	8	9	10	11	12	13	License Agreement
nome books downloa	15	16	17	18	19	20	Serial Number
	22	23	24	25	26	27	Installation Location
Labort Dealer	29	30	31	1	2	3	Installation Options
Latest Books	5	6	7	8	9	10	Pre-Installation Summary

I am going to begin by giving you good news and bad news: The good news is that if you are presently an ActionScript programmer, you will be in some familiar territory, as you already know some of the syntax you will find in this book. The bad news is that if you are presently an ActionScript programmer, you will also be in entirely new territory with the new syntax, and you will need to rethink your understanding of ActionScript. This apparent contradiction will become clear as you move through this chapter.

This chapter, as well as subsequent chapters, is going to need to do double-duty. As you will see, it is nearly impossible to talk about ActionScript without talking about Flex. As a matter of fact, ActionScript is becoming more closely associated with Flex (even though it is a separate product) than it ever was with Flash.

In this chapter, you will look at how RIAs (Rich Internet Applications) are different from traditional websites, and take a broad tour of the ActionScript 3.0/Flex 2 environment, with the details saved for subsequent chapters.

In this chapter, you will

- Explore the new ActionScript 3.0/Flex 2 environment.
- Learn what RIA is.
- Install the ActionScript 3.0/Flex 2 environment.

#### Understanding the ActionScript 3.0/ Flex 2 environment

Let's begin with a little test.

**1.** Go to a traditional HTML website like the Apress site (www.apress.com) and look for different books and authors (as shown in Figure 1-1).

Home >> Se	saich		
Search f	for books by:		
₩ Ke	ywords: pł	p	
F 158	BN:		
□ Cat	tegory: A	Categories	
E Aut	thor:	List All Authors	
E Put	blication Date: 1	999 • 01 • to 2006 • 11 •	
Search			
Your search	n has returned 47 resu	Its	
Paris 1 T	Tot 4		
safe x -	1014		
ISBN	Author	Title	eBook
1-893115-8	35-2 Gilmore, W. Jason	A Programmer's Introduction to PHP 4.0	
1-093115-5	93-3 Rischpater, Ray	Wireless Web Development with PHP and WAP	ar elos
1-893115-5	51-8 Gilmore, W. Jason	Beginning PHP 5 and MySQL: From Novice to Professional	Canada and a state
1-59059-15	50-X Lea, Chris	PHP NySQL Website Programming: Problem - Design - Solution	Caralati
1-59059-35	90-2 Andrew, Rachel	PHP Web Development with Macromedia Dreamweaver MX 2004	ar elos
1-59059-38	80-4 Zandstra, Matt	PHP 5 Objects, Patterns, and Practice	() and the
1-59059-24	8-4 Argerich, Luis	Professional PHP4	
1-59059-28	80-8 Sklar, David	Essential PHP Tools: Modules, Extensions, and Accelerators	Carallel .
1-59059-39	2-8 Darie, Cristian	Beginning PHP 5 and NySQL E-Commerce: From Novice to Professional	() and all
1-59059-50	08-4 Snyder, Chris	Pro PHP Security	() and a
1-59059-50	09-2 Stephens, Jon	PHP 5 Recipes: A Problem-Solution Approach	Chinese
1-59059-54	7-5 Gilmore, W. Jason	Beginning PHP and PostgreSQL 8: From Novice to Professional	(Caralatia

Figure 1-1. Search results on the Apress site

2. Now go to the Watergate Hotel reservation site at https://reservations. ihotelier.com/onescreen.cfm?hotelid=2560&languageid=1 (see Figure 1-2). Try out the reservation system by selecting check-in/check-out dates.

-			STREET STR	AND 1814	-	-	CONTRACTOR OF THE OWNER	COMPANY NO. INC.	CONTRACTOR OF THE OWNER WATER OF THE OWNER OWNE OWNER OWNER OWNER OWNER OWNER OWNER
	chec	k-oat o	saté, r	ooms	, adul	ts and	displayed on the calendar.	Reservation" to co	mplete your reservation.
	Octob	er 200	6 - No	vemb	er 200	16 🕨	Average Daily Rate	Check-in:	
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Deluxe Room	Check-out	
_	Ľ.,	Ľ	Ľ.,	Ľ.,	Ľ.	<u>i</u>		Nights 0	Rooms
	ŀ	10	111	12	10	14	Georgetown Suite	Adults 1	Children
8	18	17	10	15	20	21	Examples F. In	A	
2	25	24	25	20	27	28	Executive one	Taxes	
9	30	21	<u> </u>	-	-	-		Total:	
	-	1 con		1	1.	·		*First Name	"Last Name
	0	7		0	10	11		*Addness	
2	12	14	18	15	17	18		*City	"State/Province
F	20	21	22	23	24	25		*Country	"Postal Code
	27	28	29	30	1	100000		-Email	
	-	100	100	1000	2		100	"Fhone	Fax
12							Comment of the second s	1/75.4	
č	heck-						1 2 A A A A A A A A A A A A A A A A A A	Carsholder	Expiration (MM/YY)
A	dults	1	•	Room	18 1		The second s	Card Number	
1	hildra	- 1-					The second se	Arrival Information	
- is	2 and under	0	-					Comments/Requests	
		1	Reset		1		- Colorado		
		Lege	nd -c	ita to p	1010			* Fields with an asterio	R are required
A.	allable	Oate						🖌 Inslude me in futu	re email compaigns
84	demail at	Date Se Date						1	a
							12 (D)	**Einist	Reservation**

Figure 1-2. The Watergate Hotel reservation system

**3.** Finally, go to http://flexapps.macromedia.com/flex15/flexstore/flexstore. mxml?versionChecked=true and you should end up at the Flex Store, shown in Figure 1-3. Try to drag and drop items into the shopping cart and then go through the process of completing the purchase (don't worry, you're not really buying anything—this is only a demo site).



Figure 1-3. The Flex Store

Did the Friends of ED site have the look and feel of a desktop application? Or did it feel like a typical Internet site experience? Did the Watergate Hotel's reservation system start to feel a lot more like a desktop application? Finally, did the Flex Store *really* feel like a desktop application?

I chose these three sites for a reason. The first one, the friends of ED site, was a traditional HTML website. The second was built using Flash MX 2004. The last one was built using Flex, the newest of the technologies. It is the concept of the last site we will be concentrating on in this book.

### **Thinking RIA**

In order for us to get started, you will need to change your thinking a bit.

When you build a web page, you traditionally think of going from page to page. Now, let's think about what goes on here. I will use a somewhat simplified example.

Typically, you type a URL address into your browser. This is actually a request, whether you realize it or not, for the default web page of that site (typically called the **home page**). After your request is bounced all over the world by routers in microseconds, it ends up on the web server of whoever's page you are requesting. That web server then sends the requested HTML page back, once again through routers all over the world, to your web browser, where that browser reads the HTML code on the page and displays the results. If that request is for information stored in a database, the receiving web server has to pass the information on to an application server (ColdFusion, JSP, ASP, ASP.NET, or PHP), which in turn passes the information to a database server. The database server then passes the information back to the application server, which writes the HTML code with the data. The application server then passes it back to the web server which, as before, sends it back to your browser for display.

While experience has shown us that all of this technology works most of the time, and fairly quickly, there is one small problem: every time we request another page, the whole process must begin again from scratch. I think most would agree that, while it does work well, it is not terribly efficient.

What's more, I think most people can easily distinguish between an Internet application, like the friends of ED site, and a desktop application such as Microsoft Word. The whole look and feel is different (among many other differences).

Wouldn't it be nice if the whole process ran much more efficiently? And wouldn't it be even nicer if desktop and web applications had more or less the same look and feel?

To address that challenge, Macromedia (now Adobe), with the introduction of Flash MX, introduced a new term: **rich Internet application** (**RIA**). This technology, which is Flash based, overcomes many of the limitations of traditional HTML in that it is nearly indistinguishable from a desktop application.

As you may have seen in the two RIA examples earlier, the pages do not need to be rebuilt completely. Only the requested data is returned and plugged in where needed. This results in smoother and quicker responses, decreased demands on the servers, and much smaller file sizes (which lends itself nicely to mobile technology).

Also, in a traditional HTML environment, user interactivity is limited to forms and just a few buttons, and normal desktop items, such as menus, often perform poorly. The addition of these desktop tools often causes file sizes to increase dramatically, which makes for slower loading times.

As mentioned previously, with the release of Flash MX, Macromedia addressed these user concerns with a new set of programming tools, allowing developers to create RIAs to provide for greater interactivity with the user. This new set of tools overcame many of the limitations of HTML/JavaScript Internet applications. Suddenly, in an RIA environment, users could have the same interactive experience in an Internet environment that they enjoy in a desktop environment. As a bonus, this additional interactivity could be added without dramatically increasing the file size.

The release of Flash MX also saw the arrival of the first Flash server: **Flash Remoting MX**. This new server gave RIA environments a greater ability to interact quickly and smoothly with database servers as well as integrate with the Java and .NET server environments. This meant that Flash could now work as a presentation tool over a variety of programming environments. As a matter of fact, some writings have referred to Flex as a **presentation server**.

As a Flash programmer, I found this to be a welcome alternative to the less-than-ideal Java Swing classes.

Many developers, however, complained that to develop an RIA, they needed knowledge of many of the complexities of the Flash environment (timelines, scenes, and so on). To address this issue, Macromedia introduced Flex in 2004. Flex presented a more traditional programming environment without many of the complexities of Flash. As you will see throughout the book, Flex combines the ease of ColdFusion, by using a proprietary markup language called MXML, with the Java-like programming capabilities of ActionScript 3.0.

There was one further issue that needed to be addressed: state.

In traditional HTML environments, there had to be a way to remember who you are going from page to page. For instance, as you shopped for books on, say, Amazon.com, how would the site's servers remember what books you added to the shopping cart? Typically, a variety of techniques, such as cookies, session variables, etc., are used. The ability of the server to remember who you are while going from page to page is called **state**. RIA applications do not have these problems because they run on the client's machine and only access the server when needed. In other words, if Amazon.com were an RIA, the shopping cart would be located inside of the application running on your machine. At the end, when you made the purchase, all of the purchase information would be sent to Amazon.com's servers. Once again, this drastically reduces the number of times you access Amazon.com's servers and makes the whole process run a lot more efficiently.

In case you think RIA is not important, let me show you some statistics about the Watergate Hotel reservation system you tried out earlier. Since implementing the RIA system, the hotel has seen

- 89% increase in reservations
- 50% increase in revenue
- 70% decrease in call-center usage
- 50% increase in sales leads

Take numbers like that and apply them to a site like Amazon.com or eBay. The results could be staggering.

While Flash MX and MX 2004 went a long way to develop RIA, Flex takes it to a whole new level. You will be seeing why that is as you progress through this book. Here, however, let's tackle one question you may be asking yourself: "How do I build an RIA application in Flex?" The short answer: you begin with a whole new programming environment called **Flex Builder 2**. In addition, with that new programming environment, you need to stop thinking about page-to-page websites and start thinking in terms of smooth-flowing desktop-like applications.

### Flex Builder 2 and Eclipse

If you surf around on the web and look at various programming sites, you'll realize that a significant proportion of programmers use two programming environments—Java and .NET.

**Java** is a programming language introduced by Sun Microsystems in the mid-1980s and is an evolutionary step to the popular C++ environment. It is platform independent and utilized by many Internet and non-Internet applications today. (I have a toaster that uses Java. On the other end of the spectrum, the US Space Shuttle utilizes Java programming.)

The **.NET environment** was introduced by Microsoft in 2000 and is a multilanguage programming environment being utilized by many web applications today. It can employ a variety of different programming languages and compile them to a unified code on a variety of platforms.

.NET programmers mostly use Microsoft's **Visual Studio** to develop their programs. This **integrated development environment**, or IDE, employs tools to help build an application visually while it automatically writes bug-free code in the background.

A variety of IDE tools are available for Java. However, one of the most utilized ones is **Eclipse**. Eclipse is a multilanguage environment that can assist the programmer in a number of routine tasks common to all programming, such as properly formatting the code, renaming variables, and so on. You can learn more about Eclipse at www.eclipse.org.

Remember, Eclipse is not language specific. Instead, a number of members in the programming community have developed extensions or **plug-ins** to assist Eclipse with understanding programming languages like Java, C++, ColdFusion, and a variety of others. The nice part is that Eclipse, as well as most of the plug-ins available for it, is free of charge. Because of the immense popularity of Eclipse, as well as its power as a traditional programming environment, Macromedia (now Adobe) made the decision to develop the Flex IDE, Flex Builder 2, around it. This will help many programmers to develop RIAs while working in a familiar environment. Although Flex Builder is an Eclipse plug-in, sadly it is not free.

Before you can use it, you need to install it. Let me turn your attention to installing the Flex development environment.

#### **Installing Flex Builder 2**

As of this writing, Flex Builder comes in two varieties: with and without the ability to create charts. Once you have decided on which one you want, you have two ways of purchasing it: either you can download it and, within the 30-day trial period, purchase an activation key online; or you can order a boxed copy.

Flex Builder 2 comes bundled with Eclipse, Flash Player 9 (which is needed for Flex applications to run), and a single licensed version of the Flex server for testing purposes.

Once you have received your copy of Flex Builder, you need to install it. I'll walk you through the steps now.

 Start the installation process by either downloading Flex 2 from the Adobe site or inserting the disk that was shipped to you with your purchase. You'll be asked in what directory you want to save the installation files for Flex 2 (see Figure 1-4). Remember, this is not where Flex Builder 2 will be installed, but just a temporary holding place for the install files. Where you want these files is entirely up to you, but I typically like to put them in C:\Program Files\Flex 2.

🔄 Flex Builder 2 - InstallShield Wizard	$\mathbf{X}$
Location to Save Files Where would you like to save your files?	X
Please enter the folder where you want these files saved. If the fold exist, it will be created for you. To continue, click Next.	der does not
Save files in folder:	
C:\Program Files\Flex 2	
	Change
Tortal/Chield	
Back Next >	Cancel

Figure 1-4. Initial install screen

- **2.** Clicking the Next button will start some file extractions. There is nothing for you to do here.
- 3. The next screen is the opening screen for installing Flex Builder 2 and is shown in Figure 1-5. If you do not already have Eclipse installed on your computer, select the first option, Flex Builder and Flex SDK. If you are an experienced Eclipse user, you may want to select the second option. This will just extract the necessary Flex Builder files. You would then need to manually install them, using Help ➤ Software Update. Once you have made your choice, click Next.



Figure 1-5. Choosing the installation option

- **4.** The next couple of screens are just typical license agreement and introduction screens. Click Next twice.
- **5.** You will be brought to the screen shown in Figure 1-6. This will let you decide where you want Flex Builder 2 installed. I typically let it install in the default location of the Adobe folder. Specify your install location and click Next to continue.

🖫 Adobe Flex Builder 2 Installer				
	Choose Install Folder			
@ Introduction	Where Would You Like to Install?			
Choose Install Folder	C:\Program Files\Adobe\Flex Builder 2			
O Flash Player Installation	Restore Default Folder Choose			
O mealing				
O Install Complete				
InstallAnvwhere by Macrovision –				
Cancel	Previous			

Figure 1-6. Choosing the installation location

**6.** The next screen, shown in Figure 1-7, prompts you to install the Flash 9 player for each browser on your computer. Flex 2 will not work without Flash Player 9 or higher. It is important that each browser is selected.

🗟 Adobe Flex Builder 2 Installer				
	Flash Player Installation			
<ul> <li>Introduction</li> <li>License Agreement</li> <li>Choose Install Folder</li> <li>Flash Player Installation</li> <li>Previnsialization</li> </ul>	To view and debug applications developed with Flex Builder 2, you must install at least one copy of Flash Player 9. This setup can install the debug version of Flash Player in any of the following browsers.			
O Installing	Adobe Flash Player 9 for Internet Explorer			
InstallAnywhere by Macrovision - Cancel	Previous Next			

Figure 1-7. Installing the Flash Player for each browser

**7.** Click the Next button, and you are presented with the summary screen for installation (see Figure 1-8).



Figure 1-8. The confirmation screen before installation

**8.** Click the Install button, and Flex Builder will be installed along with everything else you need to develop RIA applications in Flex.

#### THE COMPLETE GUIDE TO FLEX 2 WITH ACTIONSCRIPT 3.0

If all went well, you should see a final screen telling you that everything was installed successfully, and you are now ready to start building RIAs, as shown in Figure 1-9.

🖫 Adobe Flex Builder 2 In:	italler	
	Install	Complete
<ul> <li>Introduction</li> <li>License Agreement</li> <li>Choose Install Folder</li> <li>Flash Player Installation</li> <li>Pre-Installation Summary</li> <li>Installing</li> <li>Install Complete</li> </ul>	Congratulations! Adobe Flex Builder 2 has been installed successfully to: C:\Program Files\Adobe\Flex Builder 2	
InstallAnywhere by Macrovision — Cancel	Previous	Done

Figure 1-9. Final install screen

**9.** Go ahead and click Done. You should be presented with a very strange-looking message, as shown in Figure 1-10.

ColdFusion Installation Info			
Would you like to see instructions on how to install the ColdFusion Extensions for Flex Builder 2?			
	No Yes		

**Figure 1-10.** Accessing instructions for linking ColdFusion to Flex Builder 2

Although you've now successfully installed Flex Builder 2, as this dialog box hints, you're not quite done. You'll want to install the Flex-ColdFusion extensions, and we'll get into that in the next section.

#### Installing the ColdFusion Extensions for Flex Builder 2

The primary job of Flex is to build Flash-based graphic user interfaces to present data from a variety of platforms. However, Flex has no capability of linking directly to a database. Instead, as Chapter 2 discusses in detail, you will need to use **middleware**. This is software that sets the rules for connecting with databases and handling data. Middleware could be ColdFusion, Java, .NET, or PHP. By far the easiest to use is ColdFusion (not to mention that it is also an Adobe product).

If you are going to use ColdFusion (and I highly recommend you do, since the examples in much of the later part of the book rely on it), you need to make sure that you have the latest edition (version 7.02 as of this writing) installed in order to successfully link it to Flex. Instructions for installing ColdFusion version 7.02 can be found in Appendix A in the back of this book.

As of this writing, you can download ColdFusion at www.adobe.com/ products/coldfusion/. You can go ahead and install the Flex-ColdFusion extensions if you do not have ColdFusion installed yet. You just cannot use them without version 7.02 at a minimum.

In addition to ColdFusion itself, you need the Flex-Coldfusion extensions to link Flex Builder 2 to ColdFusion. Let's assume you have version 7.02 of ColdFusion installed and want to install the extensions. When the ColdFusion Installation Info dialog box back in Figure 1-10 appears, click Yes. You will be presented with two files, as shown in Figure 1-11. One is the HTML instructions.



Figure 1-11. Files for installing the ColdFusion extensions

If you open the HTML instructions, it will give you step-by-step directions for installing these extensions on your own. I will also walk you through these instructions here if you want to install them as you work through the rest of this chapter.

Now your first instinct is to want to unzip the CF\_FB\_Extensions.zip file shown in Figure 1-11. It is not necessary.

1. Since you've already installed it, go in and fire up Flex Builder 2. You'll first see the screen shown in Figure 1-12. You will be prompted for the serial number provided with your purchase. If you don't have a number yet, just click the Try button for a 30-day trial.

Flex Builder 2 Activation
Enter your Flex Builder 2 serial number.
1234-5435-9876-3525-2978-8654
Also register charts
Charts:
⚠ Serial number is invalid.
OK Purchase Try Exit

Figure 1-12. The opening screen for Flex Builder 2

For obvious reasons, the serial number shown in Figure 1-12 is not a valid one.

**2.** If you purchased the charts option with Flex 2, you will also need to click the Also register charts check box and enter a valid serial number for that. When the correct serial numbers are entered, the boxes will gray out, and a message will confirm that the serial numbers are valid. Click OK. You should be taken to the opening screen of Flex Builder 2 (see Figure 1-13).



Figure 1-13. The opening screen for Flex Builder 2

#### INTRODUCING RIAS AND INSTALLING FLEX BUILDER 2

3. To install the extensions, just select Help ➤ Software Update ➤ Find and Install to bring up the dialog box you see in Figure 1-14 —this will allow you to install any software updates that are available. It's always a good idea to make sure that your installation is as up to date as possible.





**4.** Select the Search for new features to install option and click Next to move on to the next screen (see Figure 1-15).

🕲 Install	×
Update sites to visit Select update sites to visit while looking for new features.	
Sites to include in search:	
Eclipse.org update site	New Remote Site
	New Local Site
	New Archived Site
	Edit
	Remove
	Import sites
	Export sites
☑ Ignore features not applicable to this environment	
< Back Next >	Finish Cancel

**Figure 1-15.** Here you specify the sites on which you want to install new features.

5. From here you want to click the New Archived Site button on the right side and maneuver to the directory containing the ZIP file (see Figure 1-16). The file is in a folder located in the directory you decided to extract files into back in step 1 of the section "Installing Flex Builder 2." (I had suggested C:\Program Files\Flex 2.) Once you are in that directory, you should see a folder called Cold Fusion Extensions for Flex Builder. Go into that folder.

Select Local Site	Archive				<b>?</b> ×
Look in:	ColdFusion Ex	tensions for Flex Builder	- O I	₽▼	
My Recent Documents Desktop	CF_FB_Extens	ons			
My Documents					
My Computer					
My Network					
Places	File name:	CF_FB_Extensions		*	Open
	Files of type:	*.jar;*.zip		*	Cancel

Figure 1-16. Selecting the CF\_FB\_Extensions ZIP file

**6.** Select the ZIP file and click Open. You are presented with the dialog box shown in Figure 1-17 to do some renaming if necessary.

🕲 Edi	t Local Site 🛛 🗙
Name:	CF_FB_Extensions.zip
URL:	jar:file:C:/Program Files/Flex 2/ColdFusion Exte
	OK Cancel

Figure 1-17. The Edit Local Site dialog box

7. Do not change any names. Just click OK. This brings you back to the original screen with the CF\_FB\_Extensions.zip file selected (see Figure 1-18).

🛞 Install	X
Update sites to visit Select update sites to visit while looking for new features.	
Sites to include in search:	
에 CF_FB_Extensions.zip 이 에 Eclipse.org update site	New Remote Site New Local Site New Archived Site
	Edit Remove Import sites Export sites
☑ Ignore features not applicable to this environment	
< Back Next > F	inish Cancel

Figure 1-18. The Install dialog box with the ZIP file selected

#### THE COMPLETE GUIDE TO FLEX 2 WITH ACTIONSCRIPT 3.0

**8.** Click the Finish button. This brings you to the Updates dialog box (see Figure 1-19). Select the check box beside the ZIP file.

In Figure 1-19, I show the dialog box with the tree extended. When you select the topmost selection, the subselections get selected.

🕲 Updates	$\mathbf{X}$
Search Results Select features to install from the search result list.	
Select the features to install:	
	Deselect All More Info Properties Select Required
ColdFusion Productivity Wizards update site 1 of 1 selected. Show the latest version of a feature only Filter features included in other features on the lst	Error Detais
< Back Next > Fin	Ish Cancel

Figure 1-19. The Updates dialog box

**9.** Make sure the features are all selected, then click Next to move to the next screen, which is for accepting the license (see Figure 1-20).

🛞 Install		X
Feature License Some of the features have license with the installation.	agreements that you need to accept before proceeding	
ColdFusion Plugins 0.5.1432	ADOBE SYSTEMS INCORPORATED ADOBE FLEX BUILDER ADOBE FLEX CHARTING Software License Agreement NOTICE TO USER: THIS LICENSE AGREEMENT GOVERNS THIS AGREEMENT SHALL APPLY ONLY TO THE SOFTWAR	\$ IN: RE <sup>-</sup>
I accept the terms in the loer	ise agreement	>
O I do not accept the terms in t	the license agreements           < Back	ncel

Figure 1-20. License acceptance