Flex Solutions

Essential Techniques for Flex 2 and 3 Developers

Marco Casario

friendsof
DESIGNER TO DESIGNER™
an Apress® company

Flex Solutions: Essential Techniques for Flex 2 and 3 Developers

ISBN-13: 978-1-59059-876-4

ISBN-10: 1-59059-876-8

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

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 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit www.apress.com.

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

Credits

Lead Editors
Chris Mills, Ben Renow-Clarke

Technical Reviewer
David Williams

Editorial Board
Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Jason Gilmore, Kevin Goff, Jonathan Hassell, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager
Beth Christmas

Copy Editor
Nancy Sixsmith

Associate Production Director
Kari Brooks-Copony

Production Editor
Kelly Winquist

Compositor
Dina Quan

Artist
April Milne

Proofreader
Nancy Riddiough

Indexer
Toma Mulligan

Interior and Cover Designer
Kurt Krames

Manufacturing Director
Tom Debolski

To the memory of my grandmother, Maria.

CONTENTS AT A GLANCE

CONTENTS

ABOUT THE AUTHOR

Marco Casario is one of the most dynamic developers and consultants in the Adobe world. He has been passionate about Informatics since he was a child. Marco used to program games in Basic for the Commodore 64 before dedicating himself, while still very young, to innovative projects for the Web using Flash and Director (as far back as versions 3 and 5).

In 2001, Marco began to collaborate with Macromedia Italia. Since then he has produced and taken part in a long series of presentations, conferences, and articles, which you can find listed in detail in his blog, "Hands on Adobe World" (`http://casario.blogs.com`), which is currently receiving several thousand visitors every day.

In 2005, Marco founded Comtaste (`www.comtaste.com`), a company dedicated to exploring new frontiers in Rich Internet Applications and the convergence of the Web and the world of mobile devices. MobyMobile (`www.mobymobile.com`) and YouThru (`www.youthru.com`) are representatives of recent work.

Marco is also the founder of the biggest worldwide Flash Lite User Group (`http://groups.yahoo.com/group/FlashLite/`) and of `www.augitaly.com`, a reference point for the Italian community of Adobe users. He is Content Manager for the section dedicated to Flex (`www.augitaly.com/flexgala`).

Marco is currently busy working on the development (in Flex) of a very ambitious project concerning European bank counters. He is involved in various consulting and Flex and Flash Media Server training activities for the realization of Rich Internet Applications on behalf of buyers such as Capgemini, Engineering, IBM partners, and Adobe Systems Software Ireland Ltd.

Marco often speaks at national and international conferences such as Adobe MAX Conference, O'Reilly Web 2.0 Summit, 360Flex, FromAToWeb, AdobeLive, and many others.

ABOUT THE TECHNICAL REVIEWER

Dave Williamson is a freelance Flash Platform developer with 10 years of industry experience in the production and deployment of rich media web sites and applications. He has worked for clients such as Ford, Intel, Adidas, Microsoft, and Sony.

Dave's work has been focused on the production and integration of Flash and Flash Platform–based technologies since 1997, with a special interest in developing localized application development and (more recently) in applications for cell phones and devices.

Dave has also provided worldwide training and consultancy around Flash Platform technologies for a number of companies, including Adobe and Telefonica.

Dave's mumblings and musing can be found on his blog at `http://blog.bittube.com`.

ACKNOWLEDGMENTS

Writing a book really is a huge undertaking. During the course of the journey, you often have the feeling that you will never reach the end. Not only because you are working at night after a long day at work and on weekends, but particularly because it is a task that absorbs all your ideas and requires constant and assiduous concentration. You often get to the point of removing yourself from reality and putting personal and working relationships with the people around you to the test. After you finish writing, you realize how much patience your work colleagues, your clients, your partners, and most importantly your family had to have to not turn their backs on you. I therefore want to acknowledge certain people who are particularly important to me and helped me during the writing of this book.

Thank you to my mother, who has always believed in me and pushed me to improve myself and to see beyond the surface of things. Thank you to my father, who finally understands that it is never too late to hang out with his son. Thank you to my brother, who answered my insufferable requests during the writing of this book and who is now in Australia enjoying himself.

Thank you to my beloved honey, Katia, whose patience was really put to the test. Being near me during this period was definitely more difficult than usual, but her unconditional love, support, and help were fundamental for encouraging me to go on. Thanks for inspiring me to do my best. I love you, darling.

Thank you to my partner, Raffaele, and my colleagues Emanuele, Constantine, and Francesco, who were often abandoned by me, but who knew how to take care of their own (and often my own) work commitments. We will do great things together.

Thank you to my friends Fabrizio, Renato, Daniele, Juri, Marmotta, and Alessio, who have not left my side even after all my refusals to their invitations because of this book.

A special thanks to Chris Mills, who gave me the opportunity to write this book and whose ideas and corrections have made it better.

Finally, an enormous thank you to the people who made this book possible and without whose help and tips it could not have been a success: Ben Renow-Clarke, Beth Christmas, and the staff at friends of ED. They are all passionate people whose primary job is to publish the very best.

INTRODUCTION

Adobe Flex has revolutionized the way in which Rich Internet Applications are developed and used. The Flex framework ecosystem has expanded to include Flash Player 9, the declarative MXML language, the new and updated ActionScript 3 language, the SDK, the charting components and the LiveCycle Data Services.

So we can define the Flex framework as a component-based development framework for delivering Rich Internet Applications for the Adobe Flash Player runtime.

I have been using Flex since its inception. As a Flash developer, I quickly understood the power behind Flex because it used the same development approach I used for Java development with Eclipse or Net Beans. No more timelines, poor debugging and code editor tools, or evil movie clips ☺.

Since I first opened Flex Builder and learned the basics of Flex, I've fallen in love with it: a framework to create Flash applications easily. Thank God. The first version of Flex suffered from Macromedia's poor market positioning of the technology (at that time, it was Macromedia, not Adobe). Flex 1 required an expensive server module to compile applications. How could I convince my boss to buy a Flex server just to make my life easier for Flash application development?

Things changed with the release of Flex 2.

Thanks to the new pricing model, the community of developers, the announcement of the release of Flex 3 as open source, and the power of the framework itself, the interest of Flex technology has increased worldwide.

The time was ripe to start thinking of sharing my knowledge and experiences in Flex development. It was almost one year ago when Chris Mills, the editor-in-chief at friends of ED, met me at the Flash On The Beach conference in Brighton, England to discuss writing a book on Flex. My idea of creating a book of built-in, ready-to-use solutions was accepted enthusiastically.

This book was a long journey, not only because of the number of Flex features but also because of the announcement of Flex 3 and Adobe Integrated Runtime (AIR) by Adobe. In

fact, after working with the beta of Flex 3 and Adobe AIR, we realized that the new features and capabilities that the framework included should be covered.

So I stopped writing and updated all the solution examples to Flex 3 Beta 2. New solutions that covered Flex 3's new features were added as well as a full chapter dedicated to Adobe AIR.

The Flex framework is huge, so this book consists of the following diverse chapters:

- **Chapter 1**, "Flex Basics." This chapter discusses solutions for basic programming tasks such as data binding and the event model of Flex.
- **Chapter 2**, "Using Flex Components." This chapter discusses extending, customizing, and developing components with ActionScript 3.
- **Chapter 3**, "Working with Data Models and the Value Object." This chapter discusses how Flex manages data, how to work with complex MXML and ActionScript data, and how to convert ActionScript Classes as a data model.
- **Chapter 4**, "Validating and Formatting Data." This chapter discusses using and extending the `Validator` and `Formatter` classes to validate and format data.
- **Chapter 5**, "Managing Complex Data." This chapter discusses working with complex data using the new `Collection` classes, applying sort-and-filter operations to data, and using the `Cursor` class to move through the collection.
- **Chapter 6**, "Working with Remote Data Using the RPC Classes." This chapter discusses two-way communication with remote data using the HTTP calls to JSP, PHP, and ColdFusion pages; consuming web services; and invoking remote classes with the RemoteObject and AMF3 format.
- **Chapter 7**, "Displaying Data with List-based Components." This chapter discusses displaying data with the list-based controls available within the Flex framework, customizing the content using the item editors and item renderers, and working with the new `AdvancedDataGrid` components of Flex 3.
- **Chapter 8**, "Compiling and Deploying Flex Applications." This chapter discusses using the Flex command-line compiler to compile application and components, customizing the HTML wrapper to embed the compiled SWF file, and deploying a Flex application.
- **Chapter 9**, "Designing and Programming the Look and Feel of Flex Applications." This chapter discusses changing the look and feel of a Flex application using styles, CSS, skins, and premade Flash components.
- **Chapter 10**, "Flex Security." This chapter discusses the sandbox security of Flash Player and overcoming it with the cross-domain policy file, storing local persistent data with `SharedObject`, using `LocalConnection` to communicate between content running in Flash Player, creating a proxy server in PHP and JSP, and securing Flex applications.
- **Chapter 11**, "Advanced Flex Builder Techniques." This chapter discusses improving and customizing the Flex Builder IDE, documenting API language reference with the ASDoc tool, and improving the startup performance of an application.

- **Chapter 12**, "More Flex Framework Libraries and Utilities." This chapter discusses working with charting components, adding multimedia content to a Flex application, printing contents in Flex, and uploading files from clients to servers (and vice versa).
- **Chapter 13**, "User Navigation in Flex Applications." This chapter discusses navigating a Flex application with Navigator containers and using state and transition.
- **Chapter 14**, "Migrating Flex Applications on the Desktop with Adobe AIR." This is an entire chapter dedicated to AIR application development using Flex Builder, taking advantage of file system access, creating occasionally connected applications, and more.

Welcome to the revolution!

Who this book is for

If you want to learn more about Flex, this is the book for you. It is intended for readers who want to take their knowledge further with quick-fire solutions to common problems and best practice techniques to improve their Flex skills for Rich Internet Application development.

Moreover, this book is also aimed at readers who do not know Flex, but want to learn what they can do with Flex by using real-world examples.

Whether you are a Windows, Mac, or Linux developer, this book will work for you. Solutions and examples are intended for all platforms. Throughout the chapters you'll find detailed information that takes into account the differences between these platforms.

What you need

To follow and create the examples shown in this book you'll need the Flex Builder authoring tool. You can download a 30-day trial of Flex Builder here: `www.adobe.com/products/flex/`. For the solutions that use Flex 3, you can currently download the latest Flex Builder 3 beta from `http://labs.adobe.com/technologies/flex/`. (Once Flex 3 goes live, it will be available from the first address.)

You can also decide to use your favorite editor and compile the Flex solutions using the totally free Flex SDK that you can download from `www.adobe.com/products/flex/sdk/`. Or you can download the current beta Flex 3 SDK here: `http://labs.adobe.com/technologies/flex/sdk/`.

All the solutions and examples used in this book are downloadable from `http://flexsolutions.comtaste.com` or `www.friendsofed.com`.

The Flex Solutions site

While I was writing the book, I had one main goal in mind: to create an immortal book that can change according to inevitable updates. My solution was to create a site that will be constantly updated with new and revised content: http://flexsolutions.comtaste.com.

The site (along with www.friendsofed.com) will also be used for tracking the errata page for the book, so make sure to check back frequently for any updates.

Flex resources

Each topic in this book is presented in the context of an applied solution. Although a brief introduction exists for each solution, this book is not intended as a reference or documentation book.

The Flex documentation is huge and very well written. It covers all the aspects of Flex in a comprehensive way. You can download the complete set of Flex documentation for free from www.adobe.com/support/documentation/en/flex/.

Another must-see is the Adobe Flex Documentation Team blog, in which you'll find updates, new content, and other helpful information: http://blogs.adobe.com/flexdoc/.

You'll also find a whole load of dedicated Flex resources at the following sites:

- www.flex.org
- http://weblogs.macromedia.com/mxna/
- www.adobe.com/devnet/flex
- www.adobe.com/devnet/air
- http://onair.adobe.com/blogs/videos/
- http://flexcoders.org/
- www.adobe.com/cfusion/webforums/forum/index.cfm?forumid=60
- www.360conferences.com/360flex/
- http://flexsolutions.comtaste.com

Questions and contacts

Please direct any technical questions or comments about the book to flexsolutions@comtaste.com.

For more information about other Flex books, see our web site: www.friendsofed.com.

Common terminology

Following is a list of terms used throughout the book:

- **Adobe AIR**: The new Adobe Integrated Runtime, a cross-operating system runtime that allows developers to leverage their existing web development skills (Flash, Flex, HTML, JavaScript, Ajax) to build and deploy RIAs to the desktop.
- **Adobe Fireworks**: A bitmap and vector graphics editor.
- **Adobe Flash**: The tool used for authoring SWF files.
- **Adobe Illustrator**: A vector-based drawing program.
- **Adobe LiveCycle Data Services ES**: A server-side J2EE module that has a powerful data services architecture and programming model to synchronize data between client and server in enterprise environments. (This book does not cover this module.)
- **Adobe Photoshop**: A bitmap-based graphics editor.
- **Flash Player**: A Flex application runs within the Flash Player deployment platform.
- **Flex Builder**: The integrated development environment (IDE) for developing and compiling Flex applications.
- **Flex charting components**: This additional set of libraries provides interactive charts and graphs that enable rich data dashboards and interactive data analyses.
- **Flex SDK**: The Software Development Kit used to create Flex applications without the use of Flex Builder.
- **Rich Internet Application (RIA)**: This term, which was created by Macromedia in 2001, indicates an application that runs in a web browser and that has the features of traditional desktop applications.

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`**.

Pseudo-code and variable input are written in *`italic 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: ➥.

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

1 FLEX BASICS

Flex was created for developers who intend to create Rich Internet Applications, putting together a set of tools and technologies, and supporting standard-based languages to build and deploy scalable web applications.

Flex is based on the functionalities of Flash Player 9 runtime, which takes care of the client-side logic and enables it to also interact with the JavaScript and HTML content of the browser. The penetration rate of the Adobe player and its cross-platform nature enables you to save a lot of time that would otherwise be spent testing and debugging applications on different browser versions and different operating systems.

Flex applications are composed of MXML and ActionScript 3.0 files. The first, MXML, is based on the standard XML (Extensible Markup Language) and defines its layout from user interfaces through the use of tags that correspond to visual elements (containers and controls) and to nonvisual aspects of the application (data-binding and server-side resources.) Every tag has its respective ActionScript class, which then becomes compiled into an SWF file. ActionScript 3.0 is a programming language based on the standard ECMA Script 262 (as is JavaScript), which enables the developer to program the behavior of the application and how the user can interact with the elements of the user interface.

This first chapter introduces the basics of designing Flex applications, illustrating the base elements that make up the Flex Software Development Kit (SDK).

Solution 1-1: Changing the Flex default properties

When an application is developed using Flex, there are a few properties that define its look and feel. The developer can personalize aspects of the application by changing these properties.

Every Flex application defines the default `Application` container—this is the root element, within which everything else is defined. The `Application` container has some default layout characteristics and style properties that define the look and feel of an application (which can be modified).

In this solution you'll see how to override the default settings of the `Application` tag and how to personalize them.

What's involved

For every chapter, a single project is defined in Flex Builder, inside which all the solutions for that chapter are developed. This project will maintain the default settings. You'll create a new folder in the project (named assets), which will contain the eventual external elements (images, fonts, and so on).

The default properties of the Application container that can be personalized are the following:

Property	Description	Default Value
backgroundColor	The background color of the entire area covered by the Flash Player	0x869CA7
backgroundGradientAlphas	The opaque background	[1.0, 1.0]
backgroundGradientColors	The colors that define the gradient	[0x9CB0BA, 0x68808C]
backgroundImage	Defines a background image	No default
backgroundGradientAlphas and backgroundGradientColors styles	Define background alpha values and gradients	mx.skins.halo. ApplicationBackground
backgroundSize	Defines the dimensions of the area of the application	100%
horizontalAlign	Defines the horizontal alignment of the application	Centered
paddingBottom	Defines the amount of bottom padding the application has	24 pixels
paddingLeft	Defines the amount of left padding the application has	24 pixels
paddingRight	Defines the amount of right padding the application has	24 pixels
paddingTop	Defines the amount of top padding the application has	24 pixels

How to build it

You'll start by creating a new Flex project from Flex Builder that you will use throughout the chapter:

1. Open Flex Builder and create a new Flex project by selecting File ➤ New ➤ Flex Project. The New Flex Project Wizard opens. Create a Basic project, as shown in Figure 1-1, click Next, and name the project Chapter_1_Flex_Basics. Click Next again.

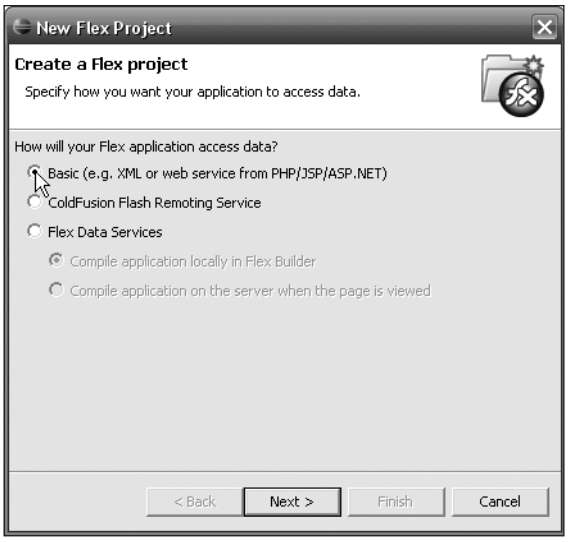

Figure 1-1. The New Flex Project Wizard guides you through the creation of a Flex project.

2. In Step 3 of the wizard, Flex Build Path, change the name of the main application file to Chapter_1_Solution_1.mxml.
3. Click Finish.

The new project will include a main application file with the following code by default:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute">
</mx:Application>
```

4. Add the backgroundColor property to define the background of the application, changing the background color:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute"
backgroundColor="#808080">
```

The background color defines a color that will be used to create a gradient effect. With the code written previously, you obtain a faded background color that starts at light gray and gradually changes to a darker tone of gray.

5. To obtain a solid color, avoiding the fading effect, set the backgroundAlpha property to 0:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute"
  backgroundColor="#808080"
  backgroundAlpha="0">
```

In this way, the background color of the Application assumes a gray color defined by the hexadecimal value in the backgroundColor property (#808080).

Another method used to create a solid color effect is to define the same two values in the backgroundGradientColors property:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute"
backgroundGradientColors="[#808080, #808080]">
```

The backgroundGradientColors property enables you to define the two color values to use for the gradient.

The backgroundImage property is used to add a background image to an application, to which you specify the path and the name of the image to use.

6. In the project, create a new folder called assets. In this folder, insert an image in one of the formats supported by Flex.

Flex takes advantage of the new Flash Player 9 runtime environment to display objects on the screen. Flash Player 9 supports a number of file formats that you can make use of in your Flex applications. You can import GIF, PNG, and JPEG files into applications.

7. Define the path and name of the background image using the backgroundImage attribute of the Application tag, like so:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute"
  backgroundGradientColors="[#808080, #808080]"
  backgroundImage="assets/bellagio.jpg">
```

Switching to Design mode in Flex Builder should already have loaded and displayed the image, as shown in Figure 1-2.

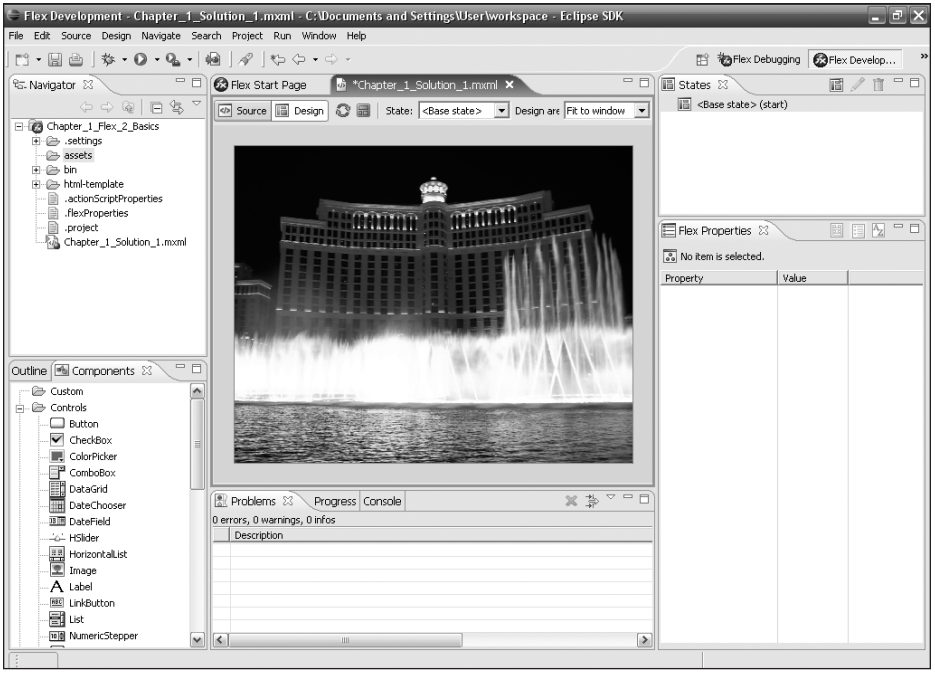

Figure 1-2. The image, imported and visualized in Design mode of Flex Builder

A little flash of color might be seen before the image is loaded, which corresponds to the defined color of the `backgroundColor` property or `backgroundGradientColors` property. There is a way to reset all the default setting defined by Flex—by setting the property `styleName` to `plain`:

- Sets the padding to 0 pixels
- Removes the default background image
- Sets the background color to white
- Left-aligns the children

It is useful to use this property when you want to overwrite all the default styles and define personalized styles:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute"
styleName="plain">
```

Expert tips

When images are imported and used as a background, the image itself will scale to fit the entire stage dimension. This might cause distortions of the image, so you should define the `width` and `height` values for the `Application` tag as the same dimensions of the image (make sure that you choose an image that is big enough for your application):