AdvancED ActionScript Components: Mastering the Flash Component Architecture

Copyright © 2006 by Antonio De Donatis

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): 978-1-59059-593-0

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
Chris Mills

Assistant Production Director
Kari Brooks-Copony

Technical Reviewers
Sas Jacobs,
Paul Barnes-Hoggett

Production Editor
Laura Cheu

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

Compositor
Dina Quan

Indexer
Becky Hornyak

Project Manager
Julie M. Smith

Proofreader
Christy Wagner

Copy Edit Manager
Nicole LeClerc

Cover Image Designer
Bruce Tang

Copy Editors
Ami Knox, Marilyn Smith

Cover Designer
Kurt Krames

Manufacturing Director
Tom Debolski
To JenFeng
# CONTENTS AT A GLANCE

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About the Author</td>
<td>xvii</td>
</tr>
<tr>
<td>About the Technical Reviewers</td>
<td>xviii</td>
</tr>
<tr>
<td>About the Cover Image Designer</td>
<td>xix</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xx</td>
</tr>
<tr>
<td>Introduction</td>
<td>xxi</td>
</tr>
</tbody>
</table>

## PART ONE  INTRODUCING THE ARCHITECTURE  1

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1  OOP in Component Design</td>
<td>3</td>
</tr>
<tr>
<td>Chapter 2  Core Classes and Component Design</td>
<td>33</td>
</tr>
<tr>
<td>Chapter 3  Exploring the UI Components</td>
<td>59</td>
</tr>
<tr>
<td>Chapter 4  Building Component-based Applications</td>
<td>91</td>
</tr>
</tbody>
</table>

## PART TWO  EXPLOITING THE ARCHITECTURE  129

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 5  Architecture-based Development</td>
<td>131</td>
</tr>
<tr>
<td>Chapter 6  XML for Defining User Interfaces</td>
<td>149</td>
</tr>
<tr>
<td>Chapter 7  Extending the Application Framework</td>
<td>171</td>
</tr>
<tr>
<td>Chapter 8  Making Your XML Life Easier</td>
<td>201</td>
</tr>
<tr>
<td>Chapter 9  The Customization Process</td>
<td>219</td>
</tr>
</tbody>
</table>
PART THREE  CUSTOMIZING THE COMPONENTS  ............... 251

Chapter 10  The Accordion Component .............................. 253
Chapter 11  The Button Component ........................................ 275
Chapter 12  The CheckBox and RadioButton Components .......... 295
Chapter 13  The List, ComboBox, and DataGrid Components .... 309
Chapter 14  The DateChooser and DateField Components ......... 349
Chapter 15  The Loader, ScrollPane, and ProgressBar Components ........................................ 371
Chapter 16  The Menu and MenuBar Components .................... 395
Chapter 17  The NumericStepper Component .......................... 427
Chapter 18  The TextArea, TextInput, and Label Components ...... 439
Chapter 19  The Tree Component ........................................... 459
Chapter 20  The Window and Alert Components ..................... 483
Chapter 21  Handling the Scrollbars .................................... 507

PART FOUR  APPENDIXES ............................................. 521

Appendix A  Locating the Source Code of the Component Architecture ........................................ 523
Appendix B  Transitions and Easing Classes ............................. 529

Index ................................................................. 535
CONTENTS

About the Author .................................................. xvii
About the Technical Reviewers ......................... xviii
About the Cover Image Designer ....................... xix
Acknowledgments ................................................ xx
Introduction ....................................................... xxi

PART ONE  INTRODUCING THE ARCHITECTURE ................. 1

Chapter 1  OOP in Component Design ..................... 3
  A very short history of the architecture ............ 4
  Creating Flash components ......................... 4
  An apparently useless component ................. 5
    Creating the Vogon component ................. 5
    Adding a method .................................. 7
  Variables, properties, and metadata tags .......... 8
    Implementing a property explicitly .......... 9
    Implementing a property implicitly .......... 11
    Properties in the authoring environment .. 12
  Inheritance ........................................... 14
    A little help from Darwin .................... 15
    Appreciating the benefits of inheritance .. 17
  ActionScript limit on multiple inheritance .... 19
  Events ............................................... 20
    What is an event in component terms? ........ 21
    Implementing a custom event ................. 21
    Triggering a custom event ................... 21
    Listening to a custom event .................. 23
    Building the example ....................... 23
  Polymorphism ....................................... 26
    And God took a rib from a Vogon ............ 26
    A method’s signature ......................... 29
    Appreciating the benefits of polymorphism .. 30
  Summary ............................................. 31
Chapter 2  **Core Classes and Component Design**  ................. 33

The legacy of the UIObject class  .................. 36
  Creating a component instance dynamically  ........ 37
    Overriding the symbolName property  .......... 38
    Overriding the symbolOwner property  .......... 38
    Overriding the className property  .......... 38
    The createClassObject method  .......... 39
  Our components join the architecture  .......... 40
Inside the process of building a component instance  .......... 44
  Step 1: Initialization  .......... 44
  Step 2: Creating the children  .......... 45
  Step 3: Drawing the component instance  .......... 45
Refining our sample components  .......... 46
The component framework  .......... 51
  The UIComponent class  .......... 51
    Accessibility/Keyboard use  .......... 51
    Other features of the UIComponent  .......... 52
  The View class  .......... 53
  The ScrollView class  .......... 54
  An ActionScript template for new components  .......... 54
Summary  .......... 57

Chapter 3  **Exploring the UI Components**  .................. 59

The Reusability Card  .......... 60
  Frequency (of use)  .......... 61
  Complexity  .......... 62
  Stability  .......... 62
  Maturity  .......... 63
  Popularity  .......... 64
Multitier applications  .......... 64
UI components provided with Flash  .......... 65
  Button components  .......... 65
    Button component  .......... 66
    CheckBox component  .......... 66
    RadioButton component  .......... 67
  Text components  .......... 67
    Label component  .......... 68
    TextInput component  .......... 68
    TextArea component  .......... 69
<table>
<thead>
<tr>
<th>Cell-structured components</th>
<th>69</th>
</tr>
</thead>
<tbody>
<tr>
<td>List component</td>
<td>70</td>
</tr>
<tr>
<td>ComboBox component</td>
<td>70</td>
</tr>
<tr>
<td>DataGrid component</td>
<td>71</td>
</tr>
<tr>
<td>Tree component</td>
<td>72</td>
</tr>
<tr>
<td>Container components</td>
<td>72</td>
</tr>
<tr>
<td>ScrollPane component</td>
<td>73</td>
</tr>
<tr>
<td>Loader component</td>
<td>74</td>
</tr>
<tr>
<td>Window component</td>
<td>74</td>
</tr>
<tr>
<td>Accordion component</td>
<td>75</td>
</tr>
<tr>
<td>Peculiar components</td>
<td>76</td>
</tr>
<tr>
<td>Alert component</td>
<td>76</td>
</tr>
<tr>
<td>DateChooser component</td>
<td>77</td>
</tr>
<tr>
<td>DateField component</td>
<td>78</td>
</tr>
<tr>
<td>Menu component</td>
<td>78</td>
</tr>
<tr>
<td>MenuBar component</td>
<td>79</td>
</tr>
<tr>
<td>NumericStepper component</td>
<td>80</td>
</tr>
<tr>
<td>ProgressBar component</td>
<td>80</td>
</tr>
<tr>
<td>UIScrollBar component</td>
<td>81</td>
</tr>
<tr>
<td>Using the UI components</td>
<td>82</td>
</tr>
<tr>
<td>A first example of interaction</td>
<td>82</td>
</tr>
<tr>
<td>Typical structure of a component</td>
<td>85</td>
</tr>
<tr>
<td>The actions layer</td>
<td>87</td>
</tr>
<tr>
<td>The assets layer</td>
<td>87</td>
</tr>
<tr>
<td>The bounding box layer</td>
<td>88</td>
</tr>
<tr>
<td>Summary</td>
<td>88</td>
</tr>
</tbody>
</table>

**Chapter 4 Building Component-based Applications**

<table>
<thead>
<tr>
<th>Screens</th>
<th>92</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building an application using screens</td>
<td>93</td>
</tr>
<tr>
<td>Content hierarchy in nested screens</td>
<td>95</td>
</tr>
<tr>
<td>Reviewing the purpose of slides and forms</td>
<td>96</td>
</tr>
<tr>
<td>Forms visibility</td>
<td>98</td>
</tr>
<tr>
<td>Conclusion: should you use slides or forms?</td>
<td>99</td>
</tr>
<tr>
<td>Screen hierarchies with external subtrees</td>
<td>99</td>
</tr>
<tr>
<td>The complete path to an external screen</td>
<td>101</td>
</tr>
<tr>
<td>Creating a slide presentation dynamically</td>
<td>102</td>
</tr>
<tr>
<td>Building the example</td>
<td>103</td>
</tr>
<tr>
<td>Importing the Slide class</td>
<td>108</td>
</tr>
<tr>
<td>Creating the screen hierarchy dynamically</td>
<td>108</td>
</tr>
<tr>
<td>Adding navigation in the master screen</td>
<td>109</td>
</tr>
<tr>
<td>Implementing the buttons-based navigation</td>
<td>110</td>
</tr>
<tr>
<td>Using Loader components in the child screens</td>
<td>111</td>
</tr>
<tr>
<td>Introducing the transitions</td>
<td>111</td>
</tr>
<tr>
<td>Importing the transition classes</td>
<td>112</td>
</tr>
<tr>
<td>Screen events and transition sequencing</td>
<td>113</td>
</tr>
<tr>
<td>Working with forms</td>
<td>114</td>
</tr>
</tbody>
</table>
Manager classes .......................................................... 114
Managing depth .......................................................... 115
  MovieClip methods for handling depth .......................... 115
  A more flexible way of stacking objects ......................... 116
  Testing the DepthManager behavior .............................. 118
Managing the keyboard focus ........................................... 120
  Defining a focus schema ............................................ 121
  Setting a default button ........................................... 123
  Tab order in a browser ............................................. 123
  Disabling the focus rect ........................................... 124
Managing windows ....................................................... 124
  A simple window-based system ..................................... 124
  Creating a window instance ....................................... 126
  Experimenting with modal windows ............................... 127
Summary ................................................................. 128

PART TWO  EXPLOITING THE ARCHITECTURE .................. 129

Chapter 5  Architecture-based Development .................... 131
  Exploiting the architecture ....................................... 132
    Key benefits of a component architecture .................... 132
    What is your job, really? ....................................... 133
  Raising the bar ..................................................... 134
    Extend, expand, and alter ....................................... 134
      Extending the architecture .................................... 134
      Expanding the architecture ................................... 136
      Altering the architecture .................................... 137
    From abstract ideas to a concrete example .................... 137
  What is an XML layout engine? .................................. 138
    Benefits of an XML layout engine ............................... 138
    Further benefits in the Flash context ......................... 140
  XLEFF .............................................................. 140
    XLEFF main features ............................................. 144
    Beyond generating user interfaces ............................. 144
    XLEFF internal architecture ................................... 145
  Summary ............................................................. 146

Chapter 6  XML for Defining User Interfaces ................... 149
  Basics of the XML data structure ............................... 150
    The Color Names section ......................................... 151
    The Styles section ................................................ 151
      Class styles ..................................................... 152
      Predefined styles ............................................... 153
      Nested styles .................................................... 153
      Custom styles .................................................... 154
    The Stage section ................................................. 155
Chapter 9  The Customization Process ........................................... 219
  Working with styles .............................................................. 220
    Parameters controlled by styles ......................................... 221
    The style lookup process .................................................. 221
    Styles as properties of a component instance ....................... 222
    The styleName property .................................................... 223
    Class-level styles ............................................................ 225
    Inheriting styles from a container .................................... 226
    Global styles ................................................................. 227
  Analyzing skins ................................................................. 228
    What is a skin? ............................................................... 228
    Handcrafted skins .......................................................... 228
    Mixed skins ................................................................. 231
    Purely coded skins .......................................................... 234
  Working with themes ............................................................ 236
    Changing skins and the mirage of code separation ................. 236
    Changing skins at authoring time ...................................... 237
    Changing skins programmatically ..................................... 240
    Skins that reflect styles .................................................. 243
    In search of a unified approach: subclassing ....................... 245
    An alternative to subclassing ............................................. 249
  Summary .......................................................................... 249

PART THREE  CUSTOMIZING THE COMPONENTS ................................. 251

Chapter 10  The Accordion Component ........................................ 253
  A minimal example ............................................................. 254
    Code-based version ......................................................... 255
    Codeless version ............................................................ 256
    XLEFF version ............................................................... 257
  The component structure .................................................... 257
    Segment header ............................................................. 258
    Segment content area ...................................................... 258
  A richer example ............................................................... 259
    Codeless version ............................................................ 259
    Code-based version .......................................................... 261
  Supported styles ............................................................... 262
    Common styles .............................................................. 263
    Specific styles .............................................................. 265
  Skinnability ................................................................. 265
    The border ................................................................. 265
    The headers ............................................................... 266
  Solved mysteries .............................................................. 268
    Inheriting styles ............................................................ 268
    Creating header styles on a per-instance basis ...................... 271
  Reasons for subclassing .................................................... 273
<table>
<thead>
<tr>
<th>Chapter 11</th>
<th>The Button Component</th>
<th>275</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal example of the Button component</td>
<td>276</td>
<td></td>
</tr>
<tr>
<td>A richer example</td>
<td>277</td>
<td></td>
</tr>
<tr>
<td>Supported styles</td>
<td>279</td>
<td></td>
</tr>
<tr>
<td>Common styles</td>
<td>280</td>
<td></td>
</tr>
<tr>
<td>Specific styles</td>
<td>281</td>
<td></td>
</tr>
<tr>
<td>Halo theme case</td>
<td>281</td>
<td></td>
</tr>
<tr>
<td>Sample theme case</td>
<td>282</td>
<td></td>
</tr>
<tr>
<td>Skinnability</td>
<td>284</td>
<td></td>
</tr>
<tr>
<td>Replacing the purely coded skin</td>
<td>285</td>
<td></td>
</tr>
<tr>
<td>The 32 skins of a button</td>
<td>289</td>
<td></td>
</tr>
<tr>
<td>Implementing a toggle button</td>
<td>289</td>
<td></td>
</tr>
<tr>
<td>Emphasizing a button instance</td>
<td>289</td>
<td></td>
</tr>
<tr>
<td>Iconic buttons</td>
<td>289</td>
<td></td>
</tr>
<tr>
<td>Solved mysteries</td>
<td>290</td>
<td></td>
</tr>
<tr>
<td>A purely coded classic: the pill button</td>
<td>290</td>
<td></td>
</tr>
<tr>
<td>Reasons for subclassing a Button component</td>
<td>293</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 12</th>
<th>The CheckBox and RadioButton Components</th>
<th>295</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal example of the CheckBox and the RadioButton components</td>
<td>296</td>
<td></td>
</tr>
<tr>
<td>XLEFF version</td>
<td>297</td>
<td></td>
</tr>
<tr>
<td>Comparing the authoring parameters</td>
<td>297</td>
<td></td>
</tr>
<tr>
<td>Supported styles</td>
<td>298</td>
<td></td>
</tr>
<tr>
<td>Common styles</td>
<td>298</td>
<td></td>
</tr>
<tr>
<td>Specific styles</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Skinnability</td>
<td>302</td>
<td></td>
</tr>
<tr>
<td>Solved mysteries</td>
<td>305</td>
<td></td>
</tr>
<tr>
<td>Where to find the RadioButtonGroup instance</td>
<td>305</td>
<td></td>
</tr>
<tr>
<td>Reasons for subclassing the CheckBox and the RadioButton components</td>
<td>307</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 13</th>
<th>The List, ComboBox, and DataGrid Components</th>
<th>309</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal example including the List, ComboBox, and DataGrid components</td>
<td>310</td>
<td></td>
</tr>
<tr>
<td>XLEFF version</td>
<td>313</td>
<td></td>
</tr>
<tr>
<td>Richer examples</td>
<td>314</td>
<td></td>
</tr>
<tr>
<td>Itemization</td>
<td>315</td>
<td></td>
</tr>
<tr>
<td>Custom labels</td>
<td>317</td>
<td></td>
</tr>
<tr>
<td>Scrolling</td>
<td>320</td>
<td></td>
</tr>
<tr>
<td>Sorting</td>
<td>323</td>
<td></td>
</tr>
<tr>
<td>Selection management</td>
<td>327</td>
<td></td>
</tr>
<tr>
<td>Making it editable</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td>Supported styles</td>
<td>333</td>
<td></td>
</tr>
<tr>
<td>Common styles</td>
<td>336</td>
<td></td>
</tr>
<tr>
<td>DataGrid-specific styles</td>
<td>336</td>
<td></td>
</tr>
<tr>
<td>List-specific styles</td>
<td>337</td>
<td></td>
</tr>
<tr>
<td>ComboBox-specific styles</td>
<td>337</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 19  The Tree Component .............................................. 459
  Minimal example of the Tree component ................................. 460
  XLEFF version of the minimal example .................................. 464
  Supported styles .................................................................. 465
    Color styles .................................................................... 466
    Text styles ...................................................................... 466
    Animation styles .............................................................. 466
    Icon styles ...................................................................... 467
    Other component-specific styles ........................................... 467
    A note on skins .................................................................. 467
    Stylizing the minimal example ............................................. 468
  Solved mysteries ................................................................... 470
    Taking full control ............................................................ 471
    Implementing isBranch and other XML attributes .................. 477
  Reasons for subclassing the Tree component ............................ 480

Chapter 20  The Window and Alert Components ............................ 483
  Minimal example of the Window and Alert components ............... 484
    Dynamically creating windows ............................................. 487
    Dynamically creating alerts ................................................. 489
  Managing the content of a Window instance ............................ 490
  Supported styles .................................................................. 495
  Skinning the Window and Alert components ............................ 499
    Skin properties of the Window component ............................ 499
    Skin properties of the Alert component ................................. 500
  Adding skins to our previous stylized example ......................... 500
  Reasons for subclassing the Window and Alert components ........ 505

Chapter 21  Handling the Scrollbars ............................................ 507
  Minimal example of the UIScrollBar component ....................... 508
  Customizing the scrollbars inside a component ......................... 510
    Step 1: Building a stylized version of the DataGrid component ... 510
    Step 2: Skinning the scrollbars ............................................ 512
  Conclusion .......................................................................... 519

PART FOUR  APPENDIXES ......................................................... 521

Appendix A  Locating the Source Code of the Component
  Architecture .................................................................... 523
  If you are a Windows user .................................................... 524
  If you are a Mac user ......................................................... 525
  FLA source files ................................................................ 525
  Link them ........................................................................ 526
Antonio De Donatis, who graduated in Computer Science from Pisa University, has been designing and implementing object-oriented software since 1989, seems like a lifetime, when the first OOP developing tools appeared on the market. So far Antonio has managed, designed, and in several cases single-handedly implemented numerous projects for a variety of industries, ranging from media/communications to pharmaceuticals.

Antonio has worked for both large corporations and leading new media agencies, and currently trades under the name of Managed Source Limited, based in Surrey, England, where he has lived since 1998 after moving from Italy, his native country.

His commercial experience with Flash goes back to the fourth version of the software, released in 1999. In recent years, the object-oriented evolution of ActionScript has allowed him to reuse knowledge and techniques that he mastered when utilizing older programming languages such as C++ and Java.

Antonio is a specialist in the design of component-based architectures for the implementation of knowledge and content management systems and is now working on several projects, including the open source XML layout engine for Flash mentioned in this book, the latest version of which can be downloaded from www.xleff.org.

Antonio considers programming a form of art and also enjoys chess, painting, and photography.
ABOUT THE TECHNICAL REVIEWERS

Sas Jacobs is a web developer who loves working with Flash. She set up her 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 ASP.NET, Flash, XML, and databases. Sas has spoken at such conferences as Flash Forward, MXDU, and FlashKit on topics relating to XML and dynamic content in Flash. She is the author of the book *Foundation XML for Flash* (friends of ED, 2005) and contributed two chapters to *Object-Oriented ActionScript for Flash 8* (friends of ED, 2006).

After studying architecture for seven years, Paul Barnes-Hoggett changed his mind and decided to spend his time designing the “intergoogle.” He spent time as a lead developer at boxnewmedia, where he built award-winning sites for clients such as Select Model Management. (In his own words, he admits, “It’s a tough job looking at pictures of beautiful people all day, but someone has to do it.”) He set up Eyefodder in 2003, which specializes in building rich Internet applications for the media industry, and has built solutions for clients including FHM, Adidas, Air Miles, and ITV. When not pushing pixels, Paul likes to eat, drink, and be merry. To get in contact with him, visit www.eyefodder.com.
**Bruce Tang** is a freelance web designer, visual programmer, and author from Hong Kong. His main creative interest is generating stunning visual effects using Flash or Processing.

Bruce has been an avid Flash user since Flash 4, when he began using Flash to create games, websites, and other multimedia content. After several years of ActionScripting, he found himself increasingly drawn toward visual programming and computational art. He likes to integrate math and physics into his work, simulating 3D and other real-life experiences onscreen. His first Flash book was published in October 2005. Bruce’s folio, featuring Flash and Processing pieces, can be found at [www.betaruce.com](http://www.betaruce.com), and his blog at [www.betaruce.com/blog](http://www.betaruce.com/blog).

The cover image uses a high-resolution Henon phase diagram generated by Bruce with Processing, which he feels is an ideal tool for such experiments. Henon is a strange attractor created by iterating through some equations to calculate the coordinates of millions of points. The points are then plotted with an assigned color.

\[
x_{n+1} = x_n \cos(a) - (y_n - x^n_p) \sin(a)
\]

\[
y_{n+1} = x_n \sin(a) + (y_n - x^n_p) \cos(a)
\]
ACKNOWLEDGMENTS

I would like to thank every member of the team involved in this project—Chris Mills for sharing the ambitious vision of this book since its inception; Julie Smith for her essential help in coordinating and controlling the disparate processes that have made this book possible; Sam Robbins, Sas Jacobs, and Paul Barnes-Hoggett for their constructive criticism that undoubtedly raised the quality of the content; my warmest thanks also go to Ami Knox, Laura Cheu, and Marilyn Smith for ensuring that my work was properly transferred in a professional format.

On a more personal note, I would like to thank my parents for their infinite love and support and my special friend Rowena Jay, talented photographer, for my portrait utilized in this book.

Finally, I cannot end this section without mentioning my sister Angela, otherwise she will not talk to me ever again.
AdvancED ActionScript Components: Mastering the Flash Component Architecture is about exploiting the Flash Component Architecture to rapidly produce powerful Flash applications by reusing the many powerful components included in it, its underlying functionality, and the knowledge captured in its source code.

In the world of software development, the concept of reusability is frequently met with skepticism or even total disbelief. Here are the most common reasons:

- Components are released most frequently in a compiled format, meaning that their implementation cannot be modified or amended whenever needed.
- Without proper guidance, the time required to learn how to reuse functionality can become comparable with the time requested to implement it from scratch.
- Experienced developers tend to trust their own code more than code written by someone else.

The first point in the previous list does not apply to the Flash Component Architecture since its source code is available with the Flash authoring environment—a truly major benefit, because developers can both learn from it and tweak it if necessary.

I wrote this book to address the second point by providing a concrete guide to teach you how to reuse each component successfully, and in a fairly convenient amount of time.

Finally, experienced developers can trust this book since it provides information that can truly save a lot of their valuable time. Being a developer myself, I know how disappointing it can be to deal with incomplete information that leaves you with time-consuming problems to solve. That is why I took particular care in providing complete examples that go beyond using the components to also demonstrate how to fully customize them, and how to fix any bugs in their default implementation.

The structure

Component-based development can be very intimidating for the uninitiated. That’s why the content has been organized into three parts:

Part One: Introducing the Architecture—A fast-paced exploration of how key object-oriented concepts are applied to the design and implementation of the Flash Component Architecture. The first four chapters show you how to build your own components based on the shared functionality made available by the component architecture and how to efficiently reuse those components within the framework of a component-based application.
Part Two: Exploiting the Architecture—Explains the most advanced programming techniques in this topic area, with the help of XML as the ideal language for describing the structure of a component-based user interface. By reading Chapters 5 to 9, you will gain the knowledge required to master the processes of utilizing and customizing any component in the architecture. This part also includes XLEFF, an XML layout engine for Flash, implemented by the author and released under an open source license. XLEFF allows dynamic generation of user interfaces from XML definitions.

Part Three: Customizing the Components—This part of the book is the largest and provides plenty of complete examples that show you how to use and customize each standard component in the architecture. Chapters 10 to 21 allow you to rapidly learn how to use and customize the components proficiently by grouping them according to their common purpose and highlighting their shared functionality.

**Intended audience**

The Flash Component Architecture is a vast topic, and this book has been designed to offer you a very efficient and effective path to learning it. However, in order to benefit from it, you should have a good understanding of ActionScript and be familiar with XML and the Flash authoring environment.

This book is also very valuable for developers who have worked with other OOP languages such as Java or C++ and are now considering the Flash technology for building rich Internet applications.

**Version issues**

Another major plus of this book is that its contents are compatible with both the Flash MX 2004 and Flash 8 authoring environments. What we call the Flash Component Architecture was originally named the Macromedia Component Architecture Version 2 when first introduced in Flash MX 2004. The same architecture, slightly improved by fixing a few bugs, has been included in the recently released Flash 8.

The techniques illustrated in this book are based on ActionScript 2, the language supported by both versions of the Flash authoring environment. All of the examples have been tested in both Flash MX 2004 and Flash 8, and any differences in operation on the two versions have been highlighted as they occur.
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.

The complete source code

This book includes numerous complete examples that can be downloaded from the web by visiting this book's page at [www.friendsofed.com](http://www.friendsofed.com).

You can also visit the website [www.xleff.org](http://www.xleff.org) to obtain the latest version of XLEFF, the open source XML layout engine discussed in this book.