

The Essential Guide to Flash CS4 AIR Development

Marco Casario with Andrew Shorten,
Koen De Weggheleire, and Matteo Ronchi



The Essential Guide to Flash CS4 AIR Development

Copyright © 2009 by Marco Casario with Andrew Shorten, Koen De Weggheleire, and Matteo Ronchi

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): 978-1-4302-1588-2

ISBN-13 (electronic): 978-1-4302-1589-9

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 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.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

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 Editors Clay Andres, Matthew Moodie	Associate Production Director Kari Brooks-Copony
Technical Reviewer Devon Wolfgang	Production Editor Katie Stence
Editorial Board Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh	Compositor Dina Quan Proofreader April Eddy Indexer Broccoli Information Management
Project Manager Beth Christmas	Artist April Milne
Copy Editor Damon Larson	Interior and Cover Designer Kurt Krames
Developmental Editor Valerie Haynes Perry	Manufacturing Director Tom Debolski

To the memory of my grandmother Maria.

CONTENTS AT A GLANCE

Foreword	xv
About the Author	xvii
Acknowledgments.....	xx
Introduction.....	xxi
Chapter 1: Introducing Adobe AIR.....	1
Chapter 2: Introducing Flash CS4	33
Chapter 3: Building Your First AIR Application.....	79
Chapter 4: Setting AIR Application Properties	125
Chapter 5: Working with the Window APIs and the Screen Class.....	159
Chapter 6: Creating Menus for Flash Applications.....	203
Chapter 7: Accessing the File System.....	283
Chapter 8: Working with the Operating System.....	353
Chapter 9: Adding Rich Media: PDF, Video, and Audio Files.....	413
Chapter 10: Working with HTML Content	479
Chapter 11: Monitoring Network Connectivity.....	517
Chapter 12: SQLite Programming in Flash CS4	551
Chapter 13: Packaging, Distributing, and Installing AIR Applications	625
Index.....	675

CONTENTS

Foreword	xv
About the Author	xvii
Acknowledgments	xx
Introduction	xxi
Chapter 1: Introducing Adobe AIR	1
Understanding the difference between browser and desktop applications	3
Installing AIR and its SDKs	4
Installing the runtime	5
International language support	7
The WebKit engine embedded in AIR	7
Installing the AIR SDKs	8
Using ADT and ADL	9
The ActionScript 3 language	9
The AIR APIs	11
Getting under the hood of AIR 1.5	13
AIR native window APIs	13
Menu classes	14
File system integration	15
Interaction with the operating system	15
Tracking user presence	16
PDF support	16
Integrating HTML content	16
Monitoring network connectivity	16
SQLite support	17
Updating installed applications	17
AIR's limitations	18
What's new in Flash with CS4	20
AIR authoring support	20
3D graphics	21
The Bone tool	22
The Motion Editor panel	23

CONTENTS

Using the preconfigured motion tweens	23
The Kuler panel	24
Improvements on sound assets.	25
The Project and Library panels	25
Integration with Flex	26
H.264 support for video in high resolution	27
Integration with ConnectNow.	27
Adobe Pixel Bender	28
Consulting the Web: Resources	29
Summary	30

Chapter 2: Introducing Flash CS4 33

The evolution of Flash CS	34
Flash CS3 vs. Flash CS4	35
Using the timeline	36
Adding effects to improve presentations.	36
The Deco tool	37
Support for Adobe Media Encoder and the new audio library.	38
The XFL format	38
Flash CS4/Flex integration.	39
Learning the new Flash IDE	39
Using the graphical panels	39
Enhancements to the Tools panel.	40
Enhancements to the library.	40
Changing panel settings	40
The stage's improvements.	41
Choosing the right color: the Adobe Kuler panel	42
Managing Kuler's colors: the Swatches	42
Using the main work panels to create animations.	43
Working with animation presets: the Motion Presets panel	43
The timeline	45
The animation types available in Flash CS3	45
The animation types available in Flash CS4	46
Creating a 3D animation.	46
The Motion Editor panel.	47
Creating 3D animations using ActionScript 3.	48
Creating animations on the Z axis.	51
Creating a 3D inspector using ActionScript 3.	52
Creating a BlendType controller	55
Using inverse kinematics	61
Adobe product integration	63
Importing graphics projects	64
Importing an Illustrator file.	65
Importing a Photoshop file	67
Importing a Fireworks file	68
Creating and managing an AIR project.	70
Packaging and deploying an AIR application	71
Creating and deploying a simple browser in AIR with Flash CS4	71
Summary	76

Chapter 3: Building Your First AIR Application	79
Creating an AIR project in Flash	81
Saving a document as a template	82
Working with the Publish Settings panel	83
Saving the AIR project	84
Designing your application's layout	85
Improvements in integration between Adobe products	85
Importing graphical assets	86
Importing Photoshop assets	86
Distributing elements across layers	88
Importing Illustrator assets	88
Importing Fireworks assets	90
Using CSS files in AIR applications	91
Publishing an AIR application	93
Managing AIR with the Application & Installer Settings panel	94
Managing digital signatures	97
The application descriptor file	100
Creating a transparent window	101
Installing AIR applications	103
Installing and uninstalling an AIR application from the AIR file	104
Distributing an AIR application with the badge installation	104
Debugging an AIR application in Flash CS4	105
Using the trace method	106
Understanding compile-time errors	108
Using the Debug tool	109
Using the Variables panel	111
Remote debugging	113
Debugging with the try...catch...finally statement	114
Creating and deploying a simple feed reader in AIR	115
Creating icons for AIR applications	122
The AIR – Icon Images panel	122
Managing icons in the application descriptor	123
Summary	123
Chapter 4: Setting AIR Application Properties	125
Setting up an AIR project	126
Managing the document classes	126
Creating a media player in AIR	128
Creating and importing classes	129
Generating variables	129
The class constructor	130
Setting up and placing buttons	130
Building and managing the application interface	130
Listening to a song	131
Managing audio files	132
Playing the song	133
Setting the id3 tags	134
Playing the song selected by the DataGrid component	134

CONTENTS

AIR application system chrome	135
Creating transparent interfaces	136
Moving the window on the desktop	137
Filtering audio files	137
Working with advanced properties	139
Adjusting the initial size and position of windows	139
Modifying the application's window	140
Specifying the installation and program menu folders	141
Distributing the application	141
Opening the AIR file	142
The application descriptor	143
Removing transparency	144
Creating an XML application descriptor without using a wizard	145
Creating an AIR application with Flex 3	147
Creating a weather widget in AIR	148
Populating the ComboBox	150
Setting up and managing the interface	151
Setting up the application descriptor	154
Summary	156
Chapter 5: Working with the Window APIs and the Screen Class	159
Understanding native window classes	161
Creating your first native window	162
Assigning the document class to the project	164
Animating a new window	167
Managing windows	171
Adding content to a new window	171
Creating a photo gallery with native window classes	173
Managing windows with added content	175
Creating a transparent window with custom chrome	177
Creating a simple text editor using native window classes	181
Adding components in an AIR project	181
Managing the ComboBox component	182
Creating the Save Document panel	184
Writing the text editor ActionScript code	185
Initializing the TextEdit application	187
Creating full-screen windows	189
Working with multiple screens	195
Managing taskbar icons	197
Managing windows without taskbar icons	198
Summary	201
Chapter 6: Creating Menus for Flash Applications	203
Introducing the menu types	205
Window menus	205
Application menus	205
System tray and dock icon menus	206

Context menus	206
Pop-up menus	206
Examining the structure and elements of a native menu	207
Creating a native menu	207
Creating submenus for a native menu	213
Creating element separators	214
Adding items to the submenu	215
Using the menuItemSelected() method	215
Displaying the code for the native submenus	216
Testing the native menus	219
Working with application and window menus	220
Creating a menu for multiple operating systems	220
Using application menus	223
Using window menus	223
Defining the menu and its functions	225
Accessing the XML file descriptor	228
Working with context menus	234
Creating context menus	235
Native context menus	238
Nonnative context menus	240
Testing the context menus	242
Managing dock and system tray icon menus	249
Assigning a menu to an application icon	249
Preparing the application icon	254
Creating an object for raster representation	254
Using the correct class type for an application icon	255
Executing the application	256
Creating and accessing pop-up menus	260
Activating pop-up menus	261
Creating pop-up menus	261
Creating native menus dynamically	267
Extending the NativeMenu class	268
Using the XMLMenu class to create dynamic menus	273
Summary	281
Chapter 7: Accessing the File System	283
The flash.filesystem package classes	284
The File class	285
The FileMode class	285
The FileStream class	285
AIR folder shortcuts	286
The user folder	286
The user documents folder	286
The application folder	286
The application storage folder	286
The desktop folder	287
The file system root	287

CONTENTS

Accessing files and folders	287
Using relative paths	288
Using native paths	288
Using system URLs	289
Browsing and selecting a file	289
Accessing the contents of a folder	295
Listing the contents of a folder	295
Listing the contents of a folder with asynchronous functions	303
Performing operations on files and folders	305
Copying files and folders	305
Moving files and folders	306
Deleting files and folders	307
Controlling deletion methods	308
Reviewing the code and executing the file	315
Accessing a file's content	320
Using the FileStream class	320
Reading and writing to a text file	322
Using the clipboard	329
Adding content to the clipboard	330
Accessing the content of the system clipboard	331
Displaying text images	335
Displaying raster content	335
Displaying and loading hypertext links	336
Executing the application	338
Performing drag-and-drop actions	340
The phases of a dragging operation	341
Starting a dragging operation	341
Dragging objects	341
Releasing the mouse	342
Dragging text documents to the TextArea	342
Opening dragged files in the application	342
Dragging files onto the desktop	347
Compiling and executing the application	350
Summary	351

Chapter 8: Working with the Operating System 353

Invoking an application	354
Accessing command-line arguments	355
Reading command-line arguments	357
Compiling and testing the commandLineArguments.fla project	362
Using the browser to invoke the AIR application	365
Setting the allowBrowserInvocation option	366
Accessing the application ID and publisher ID of an AIR application	366
The BrowserInvokeEvent object	367
Creating an AIR application that allows browser invocation	368
Creating the document class for browserInvocation.fla	368
Setting the allowBrowserInvocation option	372
Testing and packaging the browserInvocation.air application	372

Creating a Flash application that invokes AIR applications	374
Creating the document class for the browserLauncher.fla project.	375
Compiling and testing the browserLauncher.fla project	382
Local operating system integration	383
Start on login	383
Creating an AIR application with the start on login option.	384
Creating the document class for startOnLogin.fla.	385
Testing the startOnLogin.fla project	387
File associations	389
File type settings	390
Associating an AIR application with given file types	391
System notifications.	403
Creating an AIR application that sends system notifications.	404
Creating the document class for systemNotification.fla	404
Testing the systemNotification.fla project	410
Summary.	410

Chapter 9: Adding Rich Media: PDF, Video, and Audio Files 413

Detecting PDF capability	414
Creating the DetectPDF.fla project	415
Creating the document class for the DetectPDF.fla project	418
Testing the DetectPDF.fla project	422
Loading and displaying PDF files in AIR	423
Creating the document class for the LoadingPDF.fla project	424
Testing the LoadingPDF.fla project	429
Scripting PDFs.	431
Controlling PDF navigation from AIR	432
Creating the document class for the ScriptingPDF.fla project.	434
Adding JavaScript code to PDF files	442
Creating an HTML page to show a PDF file	447
Testing the ScriptingPDF.fla project	448
Working with video	449
Loading and playing video files.	449
Creating the document class for the LoadingVideo.fla project.	450
Testing the LoadingVideo.fla project	457
Working with video sound and metadata.	458
Setting the video volume	459
Accessing video metadata.	460
Testing the LoadingVideoAdvanced.fla project	463
Showing full-screen videos	464
Testing the LoadingVideoFullscreen.fla project	468
Working with sound.	469
Loading and playing audio files.	469
Creating the document class for the ControlAudioVolume.fla project	470
Testing the ControlAudioVolume.fla project	475
Summary.	476

Chapter 10: Working with HTML Content	479
Using the HTMLLoader object	480
Caching content	485
Autoscrolling HTML content	487
Horizontal and vertical scrolling with scrollH and scrollV	487
Managing the scrollbars with createRootWindow	490
Accessing ActionScript from JavaScript	493
Interacting with the HTML DOM from ActionScript	497
Creating the AIR project to access the HTML DOM	500
Registering JavaScript events from ActionScript	503
Accessing HTML history	505
Accessing and navigating the history list	505
Navigating backward and forward	505
Limitations of the HTML engine used by AIR	511
Understanding the security model in AIR	511
Summary	515
Chapter 11: Monitoring Network Connectivity	517
Detecting network connectivity changes	518
How AIR handles network connectivity changes	518
Checking connections to network resources	520
Monitoring HTTP/HTTPS connectivity	521
Monitoring socket connectivity	524
Creating a template application	528
Using the URLRequest class in AIR	534
Extra AIR URL schemes	535
The app-storage scheme	535
The app scheme	536
URLRequest properties	536
cacheResponse and useCache	537
manageCookies	537
Using the userAgent property	539
Following redirects	540
Using the authenticate property	542
Setting default URLRequest values	545
Setting URLRequest headers	546
Summary	548
Chapter 12: SQLite Programming in Flash CS4	551
Introducing SQL: SQLite and AIR integration	552
Using SQL with relational databases	553
Connecting to a database	554
The SQLConnection class	554
Synchronous connection to a database	555
Asynchronous connection to a database	562
Creating a test application	566

Using optional connection parameters	571
The openMode parameter	571
Assigning access privileges	572
Using autoCompact	572
Using pageSize	573
Applying properties and methods of the SQLConnection class	573
The properties	573
The methods	574
Using the SQLStatement class in database operations	575
Creating tables in a database	578
Starting a connection to a database with writing privileges	580
Creating a SQLStatement instance to build your SQL query	580
Defining the SQL code to execute	581
Registering the success and failure events	581
Starting the query	581
Testing the class	584
Executing SQL instructions in asynchronous mode	584
Inserting data into a database table	587
Managing SQL operation completion	589
Managing SQL insertion errors	591
Executing SQL instructions in synchronous mode	592
Testing the application	594
Accessing SQL data	595
Using the SQLResult class	595
Querying the content of the friends table in asynchronous mode	596
Querying the content of the friends table in synchronous mode	599
Splitting returned table data into pages	601
Page layout of data in asynchronous connections	602
Accessing data in a table using value objects	606
Assigning parameters to SQL queries	610
Using parameters in the address book example	611
Using transactions in AIR	615
Using methods of the SQLConnection class	616
Starting transactions with begin	617
Restoring databases with rollback	617
Confirming changes with commit	617
Writing a SQL transaction	617
Summary	622

Chapter 13: Packaging, Distributing, and Installing AIR Applications 625

Creating an AIR package	626
Using Flash to package an AIR application	627
Specifying application settings	628
The ID option	629
Window style	630
Selecting icon images	630

CONTENTS

Specifying advanced application settings	631
Associated file types	632
Initial window settings	633
The install folder and program menu folder options	634
Specifying installer settings	634
Adding resources	635
Signing AIR applications	635
Customizing the AIR application descriptor file	637
Supporting multiple languages	639
Configuring the application to allow browser invocation	640
Using a custom application descriptor file in Flash	640
Digitally signing an AIR package	641
Certificates and the AIR installation process	641
Plan early for application signing	643
Acquiring a digital certificate	644
Signing your application using Flash	647
Migrating from an existing certificate to a new certificate	647
Using an AIRI file and the ADT tool to sign applications	649
Distributing an AIR application	649
Deploying an AIR application to your website	650
Using the seamless installer badge	651
Using the standard seamless installer badge	652
Using the enhanced seamless installer badge	654
Distributing the AIR runtime on physical media or within an enterprise environment	656
The Adobe AIR Runtime Distribution License Agreement	656
Considerations for enterprise deployment	657
Installing and uninstalling an AIR application	658
Installing an AIR application using the seamless installer badge	658
Installing an AIR application by launching an AIR file	664
Installing an update to an existing AIR application	665
Launching an AIR application	666
Providing a first-run experience upon application launch	666
Uninstalling an AIR application	667
Uninstalling the AIR runtime	668
Updating an AIR application	668
Implementing a versioning strategy	669
Updating the application	671
Presenting a custom user interface for user-initiated updates	672
Summary	673

Index	675
------------------------	------------

FOREWORD

It's been over a decade since FutureSplash, the predecessor to Macromedia Flash, and then later Adobe Flash, was first released; who could have imagined back then that a simple vector-based animation tool would evolve to become a robust platform for delivering expressive content, rich Internet applications, and high-quality video across a multitude of browsers and operating systems.

Today, Flash encompasses more than just a browser plug-in and corresponding authoring tool; the Flash platform provides web, desktop, and mobile runtimes, together with frameworks, tools, servers, and services to enable designers and developers to create high-impact experiences.

Flash Player remains at the core of the platform, with each release offering new capabilities and further improving performance. Building upon the addition of a new high-performance virtual machine for executing ActionScript code in Flash Player 9, the release of Flash Player 10 adds support for 3D transformations, custom filters and effects, advanced audio processing, and GPU hardware acceleration, enabling an entirely new class of experiences not previously achievable on the Web.

As you look to deliver an engaging experience in a way that is convenient, relevant, and personal to the user, it is important to consider whether the browser is always the most appropriate access point. For example, applications deployed to the browser typically are not available when the user is offline, they are constrained within the browser chrome, they don't allow common application interactions such as drag-and-drop from the desktop, and they limit access to files and information on the user's local machine. For some applications, these limitations cause user frustration and a desire for a richer solution.

With the release of the Adobe AIR runtime, Flash designers and developers now have the opportunity to build Flash-based applications that run on the desktop, with new capabilities to extend application functionality and overcome some of the limitations experienced when running inside the web browser. Adobe AIR builds upon the features already found in Flash Player, offering, for example, unrestricted local file system access, as well as providing a version of WebKit for rendering HTML content and SQLite for the storage and retrieval of local data.

FOREWORD

Expanding your knowledge of the Flash platform to understand the new capabilities provided by the AIR runtime is crucial if you want to deliver the most engaging experiences for your clients.

The Essential Guide to Flash CS4 AIR Development will help you get to grips with the AIR-specific features in Flash Professional CS4 and show how to leverage the new AIR APIs to create desktop applications that will work consistently on Mac, Windows, and Linux operating systems.

If you need inspiration for how you can leverage Adobe AIR, I recommend you visit the Adobe AIR Marketplace (www.adobe.com/go/marketplace/), where you will find AIR applications that you can download and try for yourself. Once you've built your own AIR applications, I encourage you to promote them on the AIR Marketplace so that others can discover and enjoy using them—we certainly look forward to seeing what you create!

Andrew Shorten
Platform Evangelist, Adobe

ABOUT THE AUTHORS



Marco Casario is one of the most dynamic developers and consultants in the Adobe world.

He has been passionate about informatics since he was little more than a child and used to program games in Basic for 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, he began to collaborate with Macromedia Italy. Since that year, he has produced and headed a long series of presentations, conferences, and articles, which you can find listed in detail in his blog entitled “Hands on Adobe World” (<http://casario.blogs.com/>), which is currently receiving several thousands of visitors every day.

In 2005, Marco founded Comtaste (www.comtaste.com/en/), a company dedicated to exploring new frontiers in rich Internet applications and the convergence of the Web, the desktop, and the world of mobile devices. MobyMobile (www.mobymobile.com/) and YouThruBiz (www.youthrubiz.com/) are representative of Comtaste’s recent work.

Marco is also the founder of the biggest worldwide Flash Lite user group (<http://groups.yahoo.com/group/FlashLite/>) and of AUGItaly (www.augitaly.com/), a reference point for the Italian community of Adobe users, in which he carries out the role of content manager for the section dedicated to Flex and AIR (www.augitaly.com/flexgala/).

He’s a professional speaker who regularly speaks at international conferences like Adobe MAX, Flash on the Beach, Flex Camp, Multi-Mania, FITC, 360 | Flex, AJAXWorld, O’Reilly Web 2.0 Summit, Adobe Live, and many other local events.

His company, Comtaste, is currently busy working on the development of some very ambitious projects concerning banks and financial agencies, and various Flex and Flash Media Server training activities for the creation of rich Internet applications on behalf of companies including Accenture, HP, Capgemini Engineering, and Adobe Systems Software Ireland.

Marco is also the author of the following books: *Adobe AIR Cookbook* (O’Reilly, 2008), *AdvancED AIR Applications* (friends of ED, 2009), and *Flex Solutions: Essential Techniques for Flex 2 and 3 Developers* (friends of ED, 2007).

Contributing Authors



Andrew Shorten is a platform evangelist for Adobe and is passionate about improving the quality, richness, and value of computer-based experiences. Andrew developed web, kiosk, and mobile user interfaces for government and enterprise customers while working at Fujitsu, and has since worked for Macromedia, Microsoft, and Adobe, where he has engaged with designers, developers, web agencies, and organizations to help them deliver rich, engaging, and successful web and desktop experiences.



Koen De Weggheleire is a faculty member of the Technical University of West Flanders (HOWEST) where he teaches Flash platform solutions (Flash, Flex, and AIR) with a smile.

As the Adobe user group manager for Belgium (www.adobeusergroup.be/) and an Adobe community expert for Flash, Koen is heavily addicted to the community and inspires it through his blog at www.newmovieclip.com/, and by speaking at various events including Adobe MAX, FITC, 360 | Flex, Flashbelt, and Flash on the Beach.

He coordinates the yearly Belgian multimedia conference MultiMania (www.multi-mania.be/), where thousands of people come together to learn from industry experts and share knowledge.

Koen is a coauthor of *Foundation Flex for Developers: Data-Driven Applications with PHP, ASP.NET, ColdFusion, and LCSD* (friends of ED, 2007). He's currently working on *Advanced AIR Applications* for friends of ED, and even more AIR books from Koen are on the way.

When there is still some time left, Koen can be found at his company Happy-Banana, together with Wouter Verweirder, doing Flash platform consultancy on advanced award-winning rich Internet applications.

When Koen is not talking ActionScript, you can find him producing music, collecting goodies, eating pizza, or renovating his 100-year-old house.

Matteo Ronchi is a Comtaste consultant. His main job is development in the multimedia and web technology fields. His professional path started in 2000 as a 3D graphics designer for product design and virtual environment visualization. Soon, following his passion for games and Internet technologies, he moved from 3D to web development, mainly using Flash and Director as core technologies. In the last two years, his work has mainly focused on ActionScript and Flex programming. In 2007, he began speaking at international conferences such as Adobe Live, From A to Web, and 360 | Flex, mainly focusing on Adobe AIR development.

Fabio Bernardi has been a web developer since 2003. The passion for programming rich and friendly web applications brought him to investigate more on Flash, ActionScript, and Flex. In 2005 he joined Comtaste, where he's been involved in several international projects. He is channel manager of the Italian Flex user group FlexGala (www.augitaly.com/flexgala/), where he's an active contributor. He is an Adobe Certified Professional Flash developer and a consultant for Adobe Italy.

About the Technical Reviewer



Devon Wolfgang is an avid Flash fan(atic) who's been fiddling and tinkering with ActionScript, Flash, Flex, and AIR since the dark web ages of 1999. He recently finished a six-year contract with the US Navy, and now resides in Ireland with his beautiful wife, Deirdre, where he holds the position of senior software developer with vStream Digital Media (www.vstream.ie/). When not wildly busy, you might just find him blogging about ActionScript or involved in one of his other preoccupations such as bad movies and bad music at <http://blog.onebyonedesign.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 just because in the end you are reduced to working on it at night after a long day at work or during the weekends, but particularly because it is a task that absorbs all of your ideas and requires constant and assiduous concentration. This often brings you to the point of abstracting yourself from reality and putting personal and working relationships with the people around you to the test. It is for this reason that at the end of your work as an author, you realize how much patience your work colleagues, your clients, your partners, and most importantly, your family had to have to not turn their back on you.

I therefore want to acknowledge these people.

Thank you to my mother, who has always believed in me and has always pushed me to improve myself and to see beyond the surface of things. Thank you to my father who has finally come to understand that it is never too late to begin to hang out with his son. Thank you to my brother who for the period of the writing of the book respected all my insufferable requests.

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

Thank you Andrew, Koen, Matteo, and Fabio for your help with this book. I am honored to work with you.

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

Thank you to my friends Fabrizio, Renato, Daniele, Juri, Marmotta, and Alessio, who even after all the refusals of their invitations because of this book, have not left my side. A special thanks to Clay Andres, who gave me the opportunity to write this book and whose ideas and corrections have rendered it better. Finally, an enormous thank you to the people who made this book possible and better, and without whose help and tips it could not have been a success—namely Beth Christmas, Damon Larson, Katie Stence, and the staff behind the scene at friends of ED. They are all passionate people whose primary job is to publish the very best content.

INTRODUCTION

If you want to learn more about AIR development using Flash CS4 and ActionScript 3, this is the book for you. You'll learn the AIR APIs for creating desktop applications with these technologies, and you'll learn to build them using object-oriented programming techniques. Whether you are a Windows, Mac, or Linux developer, this book will be useful for you, since Adobe AIR is a cross-platform desktop runtime, and the examples in this book are intended for all platforms. Throughout the chapters, you'll find detailed information that takes into account the differences between the different platforms.

The Essential Flash CS4 AIR Development site

I've created a companion site from which you can download all the code of the chapters and revised content: <http://flashcs4air.comtaste.com/>. The site will also be used for tracking the errata page for the book, so make sure to check back frequently for any updates.

What you need

In order to follow and create the examples shown in this book, you'll need the Flash CS4 authoring tool. You can download a 30-day trial here: www.adobe.com/products/flash/. You'll also need to install the AIR 1.5 runtime from www.adobe.com/go/air. Finally, all the examples used in this book are downloadable from <http://flashcs4air.comtaste.com/>.

AIR 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 official Adobe AIR documentation is huge and very well written. It covers all the aspects of official Adobe AIR in a comprehensive way. You can download the complete official Adobe AIR documentation for free at <http://www.adobe.com/support/documentation/en/air/>.

INTRODUCTION

Another important resource is the Adobe Flex Doc Team blog, where you'll find updates, new content, and other helpful information: <http://blogs.adobe.com/flexdoc/>.

Here is a list of useful resources dedicated to AIR:

- Adobe AIR Market place: <http://www.adobe.com/go/marketplace>
- <http://feeds.adobe.com>
- Adobe AIR Developer Center Feed: http://rss.adobe.com/developer_center_air_tutorials.rss?locale=en
- <http://adobe.com/devnet/air/>
- <http://onair.adobe.com/blogs/videos/>
- <http://tech.groups.yahoo.com/group/flexcoders/>
- <http://InsideRIA.com>
- <http://casario.blogs.com>
- <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 the friends of ED website, at www.friendsofed.com/.

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**.

Pseudocode 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.

CHAPTER 1

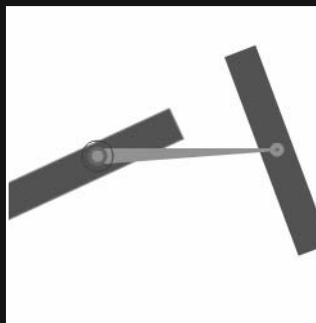
INTRODUCING ADOBE AIR

Sun **Desktop Keeley**
Desktop Keeley provides up to date sport info, showbiz gossip and news direct to your desktop.

By ashorten 30-Jul-08 ★ 7

 **TweetDeck**
TweetDeck evolves the existence of Twitter by breaking it down into a desktop application.

By Iain Dodsworth 24-Jul-08
★★★★★



 **Share Your Screen**

Meet live over the web and share your screen with up to 3 people

Collaborate with remote participants using integrated audio, chat, video and

In 2007, Adobe launched Apollo, the code name for a cross-operating system runtime that allows developers to leverage their existing web development skills to build and deploy rich Internet applications (RIAs) to the desktop.

With this runtime, web developers can leave the browser and its limitations, dictated by sandbox security, and add native functions of desktop applications that communicate and interact with Windows, Mac OS X, and Linux operating systems. The final version of Adobe's technology became AIR (Adobe Integrated Runtime).

Following are the ways in which AIR has added value over its natural competitor JRE (Java Runtime Environment):

- You can use it to create applications from scratch easily.
- You can port web applications to the desktop.
- You can deploy on different operating systems.

To create a desktop application with AIR, designers and developers don't need to learn a new language or development environment. They can create desktop applications using the same knowledge they use every day for web applications. Whether they use HTML, CSS, JavaScript, Ajax, ActionScript, or MXML, Adobe AIR can accommodate them and display the application on the desktop. And that's why learning another IDE (integrated development environment) isn't required. You can use Dreamweaver, Flash, Flex, Eclipse, Aptana, or any other environment that can use these languages. This also means that bringing a web application developed with Ajax or ActionScript (Flash or Flex) onto the desktop requires minimum effort apart from adding functions, which are provided by AIR SDKs (software development kits) to interact with the operating system.

Sun's "write once, run anywhere" motto has often shown its weakness in deploying desktop applications made with Java on different operating systems. Too many times, developers have had to rewrite entire pieces of code or find workarounds to make Java desktop applications work on Windows, Linux, or Mac OS X. With AIR, the time spent porting the application to different operating system will be null.

And if you don't believe what I'm telling you, you can trust Adobe's guarantee, which has already proved it can make any SWF (Shockwave Flash) application visible on Windows, Linux, Mac, and Sun Solaris. Check it out yourself at www.adobe.com/products/flashplayer/.

With the recent release of Flash CS4, Adobe provides an even more efficient tool to write ActionScript 3 code that can be exported natively not only as a web application in SWF format, but also as a desktop application in AIR format.

Congratulations on your choice to create a new generation of desktop applications with Flash CS4 and AIR. In this chapter, you'll learn the difference between developing a web and a desktop application using the AIR runtime.

Understanding the difference between browser and desktop applications

1

Browser or web applications are programs that work with an active Internet connection and that are loaded in a web browser. Facebook (www.facebook.com/) is an example of a well-known web application.

Desktop applications, on the other hand, are launched directly from a local computer, and don't necessarily require an Internet connection. OpenOffice Writer (www.openoffice.org/) is an example of a desktop application.

A few application features for the browser and the desktop are compared in Table 1-1.

Table 1-1. Comparison between a browser application's characteristics with those of a desktop application

Feature	Browser Apps	Desktop Apps
Internet	Required	Not necessary
Installation	None	Must be installed on the local system
Application management	Easy update	Update manager APIs
OS support	Multiple platforms	Multiple platforms
I/O support	None	File system access
Access to OS	Limited	Data stored locally via file access

As you can guess from this table, desktop applications can have more advanced functions that interact and integrate better with the local system than web applications.

Desktop and web applications are both widely used every day, so you can see how it would be advantageous to combine the features of the two. Doing so creates a single environment for developing hybrid applications. An early example of this was BuzzWord, a formerly web-only word-processing tool later released as an AIR desktop application (originally created by Virtual Ubiquity, BuzzWord was recently acquired by Adobe). The BuzzWord desktop application (shown in Figure 1-1) allows you to open and edit documents locally. Visit <https://buzzword.acrobat.com/> to learn more about BuzzWord.

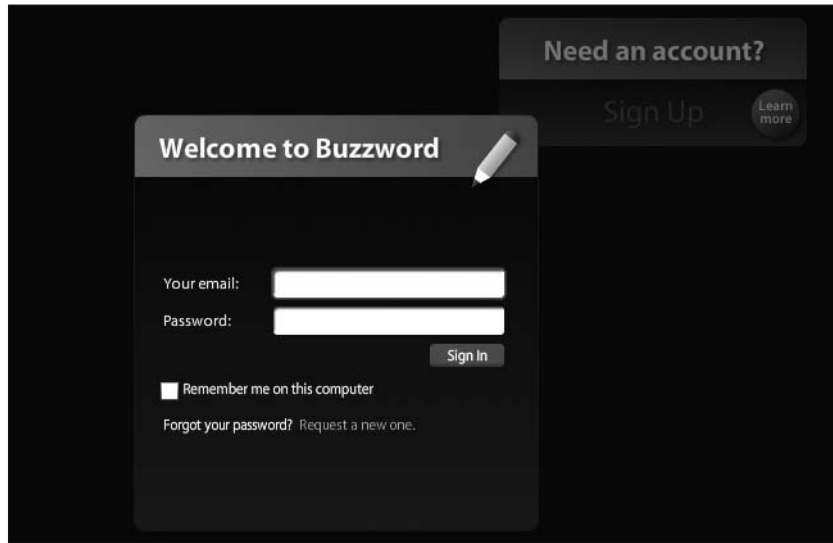


Figure 1-1. You can download BuzzWord from the Acrobat website.

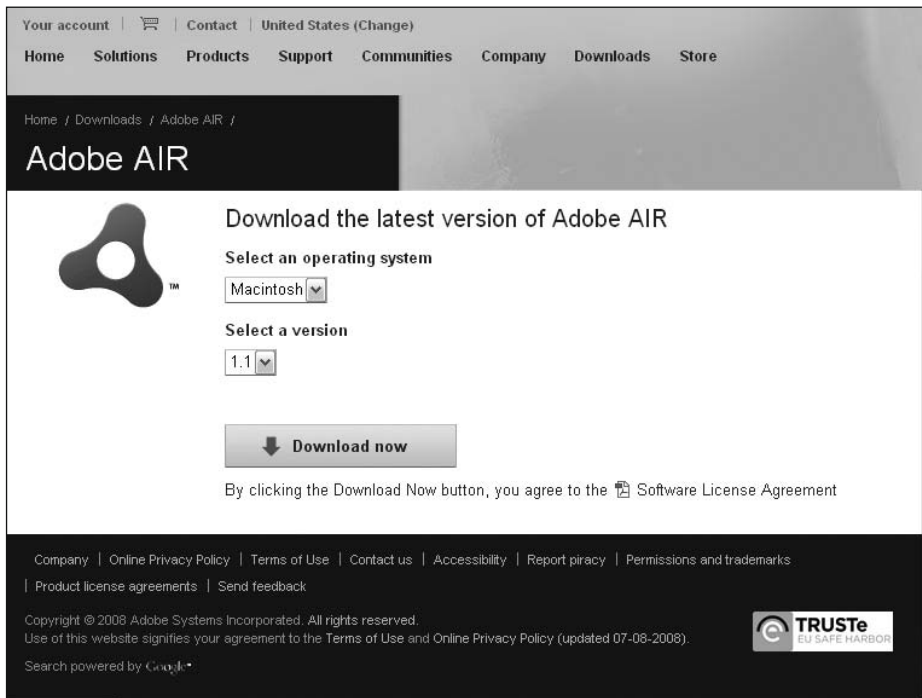
With technologies like AIR, creating desktop applications that can interact with the operating system as well as easily access remote data is quite easy and fast. Maybe the era of “occasionally connected applications” has really arrived.

Occasionally Connected Computing (OCC) is a term coined by Macromedia and used to describe some aspects of a web application when not connected to the Internet. This is sometimes a feature of a rich Internet application. (Read more on Wikipedia at http://en.wikipedia.org/wiki/Occasionally_Connected_Computing).

In the next section of the chapter, you’ll learn what’s behind the AIR runtime and how to install the SDK to start developing applications.

Installing AIR and its SDKs

To be able to install and use AIR applications, you’ll need the AIR runtime installed locally. The AIR runtime is free, and for Windows, you can download it from Adobe at <http://get.adobe.com/air/>. For other systems, you can download the runtime from <http://get.adobe.com/air/otherversions/>. Figure 1-2 shows this download page on the Adobe site.



1

Figure 1-2. Download and install the AIR runtime for your system.

Installing the runtime

Once you've downloaded the runtime, you can install it on your machine using the installation wizard. After you've done this, you'll be able to install and execute AIR applications.

Also on the Adobe website, you'll find the Adobe AIR Marketplace (shown in Figure 1-3) at the following address: http://www.adobe.com/cfusion/exchange/index.cfm?event=productHome&exc=24&loc=en_us. Here you'll find a list of applications you can download and install to help you understand what you can actually do (and what has already been done) with AIR technology.

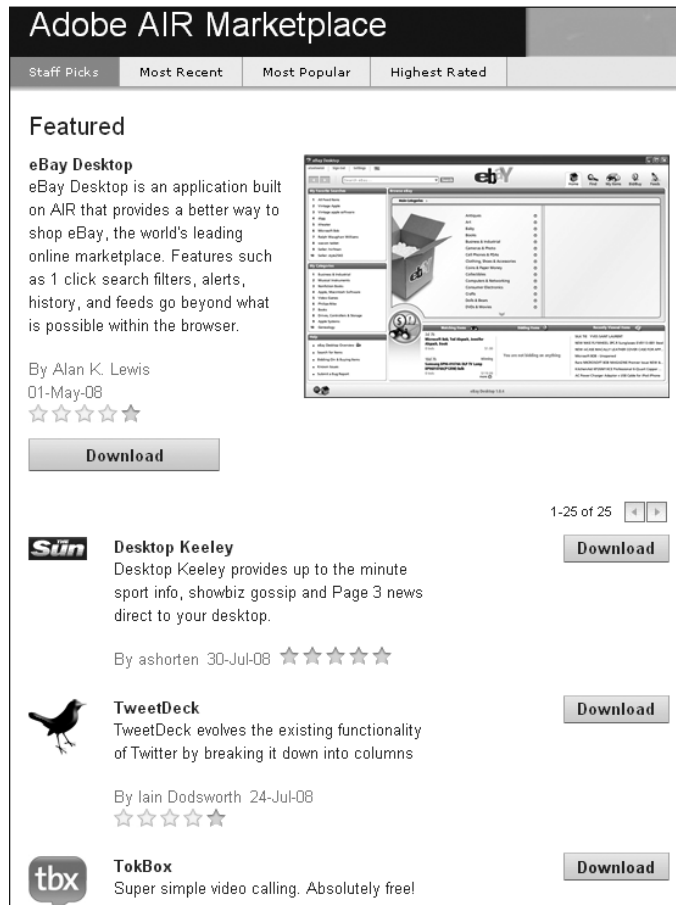


Figure 1-3. You'll find many applications for inspiration in the Adobe AIR Marketplace.

Before you install AIR, I recommend that you can meet the following system requirements:

- For Windows systems
 - Intel Pentium 2GHz or faster processor
 - Windows Vista Home Premium, Business, Ultimate, or Enterprise; Windows XP with Service Pack 2 or Windows 2000 with Service Pack 4; 512MB of RAM; 32MB of VRAM
- For Mac OS X systems
 - PowerPC G4 1.8GHz or faster processor, or Intel Core Duo 1.33GHz or faster processor
 - Mac OS X v.10.4.9 or later, or 10.5.1 (Intel or PowerPC; Intel processor required for H.264 video)
 - 512MB of RAM; 32MB of VRAM

International language support

Installing an application can intimidate many users. Fortunately, AIR installation and other runtime dialog boxes have been translated into the following languages:

- Chinese traditional
- Chinese simplified
- English
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Russian
- Swedish
- Dutch
- Czech
- Polish
- Turkish
- Spanish

The language used for the installation wizard will be taken from the language used by the local operating system. This means that if you have Windows installed in English, the AIR installation wizard will show English labels. Moreover, you can develop your AIR application to support the following:

- Building internationalized applications, including keyboard input for double-byte languages
- Localizing the name and description attributes in the application descriptor file
- Localizing error messages, such as `SQLException.detailID` and `SQLException.detailArguments`, in the SQLite database
- Obtaining an array of preferred user interface languages as set by the operating system using the `Capabilities.languages` property

The WebKit engine embedded in AIR

The AIR runtime allows you to load the three following types of formats and contents through its HTML engine:

- HTML
- JavaScript
- CSS

This is possible thanks to the WebKit engine, which is part of the WebKit Open Source Project (<http://webkit.org/>). This project is the same one used for the Apple WebKit Framework for Mac OS X and Google Chrome.

AIR uses the code base directly from the WebKit project, not from Apple's WebKit Framework.