

First Steps in Flex

Bruce Eckel

President, MindView Inc.

James Ward

Flex Evangelist, Adobe Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals. Flex and Flash are trademarks of Adobe, Inc. Java is a trademark of Sun Microsystems, Inc. All other product names and company names mentioned herein are the property of their respective owners.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact MindView, Inc. at the address at the bottom of this page.

Visit us on the Web: www.mindviewinc.com

Cover design and interior design by Daniel Will-Harris, www.Will-Harris.com

Copyright © 2008 by MindView, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

MindView, Inc.
5343 Valle Vista
La Mesa, CA 91941
Email: Wayne.MindView@gmail.com

ISBN 978-0-9818725-0-6

Text printed in the United States.

First printing, August 2008

Contents

Introduction	5
Hello, World	7
Events	13
ActionScript.....	17
Desktop Applications	29
Containers.....	33
Components	39
Data and Collections.....	47
Item Renderers.....	55
View States	59
Effects	63
Forms	69
Formatters.....	73
Drag & Drop	75
PopUps	77
Graphics	81
Styling and Skinning.....	85
Usability	93
Flash Player APIs	97
Data Visualization.....	101
Web Services.....	107
Remoting and Messaging with Java	113
Architectural Patterns	123
A: Development Tools.....	133
B: Deployment.....	137

Want Flex hands-on learning?
Look for the next Flex Jam on www.MindView.net

Introduction

We're not going to try to convince you to use Flex.

We wanted to make a small book, so we don't have the space. We'll just assume that you already want to learn Flex, and that you've had some exposure to programming.

We've found (by running hands-on *Flex Jams*—see www.MindView.net) that the best way to learn is to immediately wade in and get your ankles wet. But we don't want to throw you into the deep end where you'll get overwhelmed.

We're trying to give you just enough information, and just the right information, to get you started. Enough so that you feel confident in taking your own steps once you finish the book.

Look at the picture on the cover. It's a scary bridge, and a long way down (That's James on the bridge, in the Italian Dolomites). Once you take the first few steps, though, you'll have the confidence to cross the rest of the bridge.

As a bonus, you can find free screencasts of Bruce and James covering every chapter in this book at www.FirstStepsInFlex.com.

An important goal for this book is that it be a tool that companies can easily buy in bulk, to start everyone on their first steps to learning Flex. It also makes a nice gift at conferences. Contact us at www.MindView.net for details.

Hello, World

Build your first Flex applications.

First, download and install the 60-day free trial of Flex Builder from http://www.adobe.com/go/flex_trial. The easiest way to do this is to get the standalone installation (even if you already have Eclipse installed for some other purpose). There is also an Eclipse plugin if you are so inclined. **Note:** We do not provide downloadable source code for this book, because the examples are small and you learn a lot by writing them within Flex Builder.

Display a Label

- Select **File|New|Other**. This brings up the “Select a wizard” dialog box.
- Open “Flex Builder” and select “Flex Project.” This brings up the “New Flex Project” Dialog box.
- Type in a project name: **helloWorld**.
- Use the default location (which will already be checked).
- Select “Web application (runs in Flash Player).”
- Leave everything else alone and press “Finish.”

Your project will open in the MXML code editor. You will see the file titled **helloWorld.mxml**. Note that this is a valid XML file.

- Now add an XML tag in between the Application tags:
`<mx:Label text="Hello, world"/>`

This inserts a **Label** component into your Flex application, which should now look like the following:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application name="helloWorld"
  xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute">
  <mx:Label text="Hello, world"/>
</mx:Application>
```

Flex Builder automatically puts in the first line that you see in the listing above; this is the *doctype* line which specifies that this is a standard XML file. However, Flex ignores this line when compiling the program and so we shall leave off the redundant doctype declaration throughout the rest of the book.

***Hint:** You can easily access the ActionScript documentation for a given object in Flex Builder by pressing Shift-F2 when your cursor is on a component.*

The **name** property in the **Application** tag is *not* inserted by Flex Builder and is ignored by Flex; we use it to give the name of the file containing the program. In this case, the full file name is **helloWorld.mxml**.

Now run the application. Go to the **Run** menu and select “Run.” (If you poke around in Flex Builder you’ll find other shortcuts to running applications). This should launch your web browser and run the application within the browser. You’ll see the words “Hello, world” in a field of blue.

Note that “Hello, world” is in the upper left corner. Modify the Application tag by removing the **layout="absolute"** property and re-run the

application. You'll see that "Hello, world" is now centered.

In most of the examples in this book we will remove the **layout** property. In the *Containers* chapter we'll go into more detail about what that property does.

Note: MXML allows you to describe and configure components in a high-level, declarative fashion. However, MXML is always translated into the ActionScript language, which is then compiled into a file with an extension of **.swf**, for "Shockwave Flash." These SWF files contain bytecode that is executed by the ActionScript Virtual Machine (AVM).

Data Binding

Create a new application called **helloWorld2**, and remove the **layout** property. Place your cursor within the **Application** body and add a **String** and a **Label** component. Note that when you enter '<' and then start typing, for example, "String," Flex Builder will perform command completion for you. Make your example look like this:

```
<mx:Application name="helloWorld2"
  xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:String id="message">Hello, world</mx:String>
  <mx:Label text="{message}"/>
</mx:Application>
```

The **String** object¹ has an **id** property. The **id** is the name of the object, and is necessary so that other objects can talk to it. You'll use **id** a lot.

Here, the **id** is **message**, and the **Label** component uses this **id**, but within curly braces. Curly braces have a special meaning, which is "bind to this other object." In this case, the **text** property in the **Label** object is fetched from the **String** object. That's data

¹ If you don't know what objects and classes are, read *Introduction to Objects*, which you'll find at www.FirstStepsInFlex.com.

binding—the data in one object is bound to the data in another.

In the above example the string doesn't change during the execution of the program. Data binding is especially interesting when the bound-to data does change, because the **Label** will be automatically updated. Let's look at a more complex example:

```
<mx:Application name="helloWorld3"
  xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:TextInput id="message" text="Hello, world"/>
  <mx:Label text="{message.text}"/>
</mx:Application>
```

Now we've added another component, a **TextInput** field. The **Label's text** field is bound to **message.text**, so when you modify the **TextInput** the label will automatically change.

Under the covers, data binding generates fairly complex code to watch for changes and respond to them. But you don't have to think about this; all you need is the curly braces. This is one of many places where Flex does a lot of work for you to keep your life simple.

Further Learning

Switch to *Design View*: in the upper left corner you'll see buttons that allow you to toggle between "source" and "design." Experiment by adding components to your layout by dragging and dropping them, then switch back to "source" view and notice that you've just added more MXML components. In "design" view, try configuring the components using the "Flex Properties" pane. Also explore the different views within the "Flex Properties" pane by clicking the icons in the upper right corner of that pane.

Note: Whenever you see the '➔' symbol in this book, it means you should go to <http://www.FirstStepsInFlex.com> and type the

word to the right of the '➔' into the search box there to find out more information about that topic.

There are a number of online communities out there to help answer your questions about Flex:

- Flexcoders Yahoo Group: ➔*Flexcoders*
- Adobe Forums: ➔*AdobeForums*

Blogs are also a great way to continue learning about Flex. <http://feeds.adobe.com> has a Flex category which makes finding Flex blogs easy.