CONTENTS INCLUDE:

Getting Started with Adobe® Flash® Builder™ 4

By Eric Daugherty

ABOUT FLASH BUILDER 4

Adobe Flash Builder, part of the Adobe Flash Platform, is a multimedia platform that utilizes browser plug-ins to deliver rich content via the web. With a market penetration over 99% in most developed countries, it is a ubiquitous platform for developing rich content. Adobe Flex is an open-source Software Development Kit (SDK) used to develop rich applications on the Flash platform. Flex applications can also be delivered as applications that are installed locally using Adobe AIR®.

Flash Builder 4 is an Integrated Development Environment (IDE) from Adobe Systems Inc. for ActionScript® and Flex development. Flash Builder is built on top of Eclipse, an open source extensible development platform, and a popular Java IDE. Because it is built on Eclipse, Flash Builder 4 inherits an impressive list of plug-ins and is a familiar tool for many developers.

Hot Tip

Flex® Builder™ is now Flash Builder

The previous version of Flash Builder 4 was Flex Builder 3. Adobe changed the name for this release to avoid confusion between the open-source Flex framework and Flash Builder 4, which is a commercial IDE.

This Refcard outlines how to use Flash Builder to develop Flex applications, including creating new projects in Flash Builder, Developing, Building (compiling), Testing, and Debugging Flex applications.

Flash Builder 4 is supported on Windows and Mac OS X, and can be installed as an Eclipse plug-in or as a standalone install.

Hot Tip

Coexist

Flex Builder 3 and Flash Builder 4 can coexist on the same system. However, they cannot coexist as plug-ins in the same Eclipse install. If you have Flex Builder 3 installed as an Eclipse plug-in, you should uninstall it before installing the Flash Builder 4 plug-in, or install Flash Builder 4 standalone.

Flash Builder 4 is a commercial product and must be purchased from Adobe (a 60-day free trial is available).Flash Builder 4 pricing will be consistent with Flex Builder 3, with the Standard Edition available for $249 (USD) or $99 (USD) for an upgrade. The Professional Edition available for $699 (USD) or $299 (USD) for an upgrade. The Standard Edition is available free for Education, and Volume and Government pricing are also available.

For additional information on the differences between Flex Builder 3 Standard and Professional, see Adobe's Reasons to Buy Flex Builder 3: http://www.adobe.com/products/flex/updrade

Getting Flash Builder 4

Go to the Adobe download site - http://www.adobe.com/go/try_flashbuilder
Choose the download for your operating system from either the Standalone Installer or Plug-in for Eclipse sections.

For either the Standalone or Plug-in versions, the downloaded file is an installer. For Windows, simply run the downloaded file (.exe). For OS X, mount the DMG file and run the installer app inside the disk image. The installers will launch a wizard with instructions on how to complete the installation.

Once installed, the standalone Flash Builder can be run from the installed location (Program Files on Windows or Applications on OS X), or the plug-in version can be run by launching Eclipse.

When Flash Builder is first launched, it prompts the user for a serial number, or you can elect to start a 60-day trial. During the trial, it will display a dialog box that allows you to enter a serial number or continue with the trial each time it is launched.

Hot Tip

Applications created by Adobe Flash Builder can be published in the browser using the Adobe Flash Player, and outside the browser using Adobe AIR.

CREATE A PROJECT

Flash Builder uses projects as the parent container for all development activities. Before you start working, you must first create a new project.

Create a new project using the menu: File -> New -> Flex Project. This starts the New Flex Project wizard.

The first step requires that you specify a name for your project. You can also override the default location. There are several other important choices on this first page:

Application Type

Flash Builder supports two applications types, Web and Desktop. Web projects create Flash .swf files that can be distributed via a web server and run within a client browser. Desktop projects create .air files that can be downloaded and installed as local applications.

Hot Tip

Sandbox

Applications that run within the browser are limited to a local sandbox. They do not have access to the local computer's file system and have limited network access. AIR applications have full local file system and network access.

Server Technology

Flash Builder 4 supports integration with several server technologies, including ASP.Net, ColdFusion, J2EE, and PHP. Selecting one of these options allows Flash Builder to provide automatic configuration for remote calls. If an option other than None/Other is selected, additional information will be collected in step two. The specific data needed is dependent on the selected technology. Flex can also interact with REST and SOAP services written in any technology.

The additional steps in the wizard allow for further customization of the project, but the default values are acceptable for most beginner projects.

Hot Tip

Component Sets

The Flex SDK provides a rich set of components useful for developing applications, including everything from Buttons and Labels to rich charting components. The components in Flex 3 are called the MX components, while Flex 4 introduces the Spark components. Flex 4 projects will use a combination of these components. Flash Builder 4 continues to work with the Flex 3 SDK and MX components.

DEVELOPING

When a project is created or opened, Flash Builder displays the main window, called the workbench.

Workbench

The workbench consists of a set of Views, Editors, and Toolbars. Views provide visual access to some part of your project. The initial views include the Package Explorer, Outline, Problems, and several others. Editors allow you to edit the source files, and can be text based or visual. Toolbars provide easy access to commonly used menu items.

Perspectives can be used to manage your workbench. A perspective is a collection of views and toolbars appropriate for a certain activity. Flash Builder provides two default perspectives, Flash and Flash Debug. These perspectives open the appropriate Views and Toolbars for Development (Flash) or Debugging (Flash Debug). You can modify these perspectives or define your own to customize the Flash Builder environment to your individual taste.

Hot Tip

Getting Started with Eclipse

Flash Builder 4 is built on Eclipse, and inherits many of its features and shortcuts. For more information on Eclipse, please see the Getting Started with Eclipse Refcard. http://refcardz.dzone.com/refcardz/getting-started-eclipse

Building Flex Applications

Flex applications consist of two different types of source files. MXML files (.mxml) are XML files that allow you to arrange visual components, as well as include non-visual components and ActionScript code. ActionScript files (.as) contain ActionScript code (classes, interfaces, etc.). You can achieve the same functionality with either type of file, but they are tailored to different usages.

MXML files are ideal for visual components that contain small amounts of ActionScript code. ActionScript files are ideal for non-visual code, including model or service classes.

Flash Builder provides powerful editors for both types of source files.

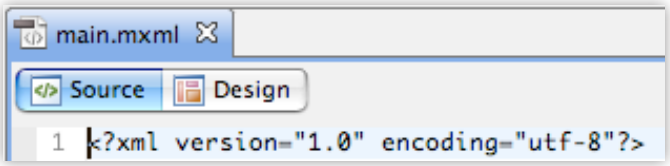
Hot Tip

Adobe Flash Catalyst

Adobe Flash Catalyst is a tool that allows designers to easily create Flex applications from artwork created from the Adobe Creative Suite. Flash Builder provides tools to import Adobe Flash Catalyst projects. For more information on Catalyst: https://www.adobe.com/go/catalyst

MXML

MXML files can be edited in two modes, Source and Design. You can switch back and forth between these modes while editing an MXML source file. This makes it easy to manage the visual layout using the designer while switching to the source view to add ActionScript and non-visual components.

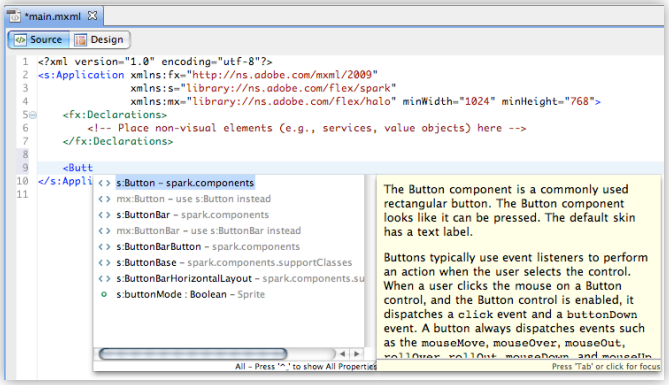


Source mode provides an XML editor with robust code completion, syntax highlighting, and error highlighting. The figure below shows the MXML editor in source mode with the code completion pop-up open.

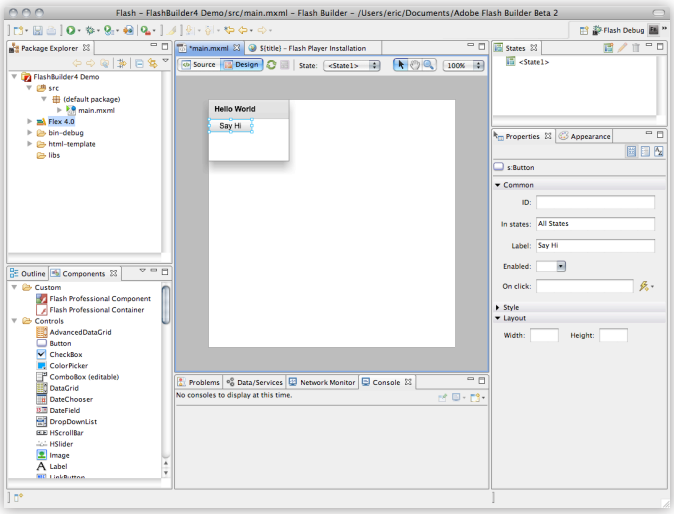
Hot Tip

ASDoc Tool Tips

You can quickly view the ActionScript documentation as a ToolTip in the MXML Source Editor and ActionScript Editor by hovering the mouse over a Flex SDK class or method.



Design mode provides a visual preview of your application. This is very useful for laying out the components, or simply checking to see how they will look without running the application. The Design mode features a list of available components in the bottom left window. You can drag and drop these onto your application to add them and adjust the layout.



Each Flex container uses a specific layout to determine the size and position of the visual components. The Flex SDK provides four general-purpose layouts. Components can be configured using absolute or relative positioning. BasicLayout allows absolution positioning, where components are assigned specific x and y coordinates. This mode allows pixel level control over the appearance of the application. However, it also has several drawbacks. When using BasicLayout, users cannot (usefully) resize the application. It can also be very difficult to internationalize an application, as buttons and labels are often different sizes based on the language. VerticalLayout, HorizontalLayout, and TileLayout all provide different approaches to layout components using relative positioning.

Name	Positioning	Description
BasicLayout	Absolute	Components are positioned using x and y coordinates.
VerticalLayout	Relative	Components are positioned in a column.
HorizontalLayout	Relative	Components are positioned in a row.
TileLayout	Relative	Components are positioned in a grid.

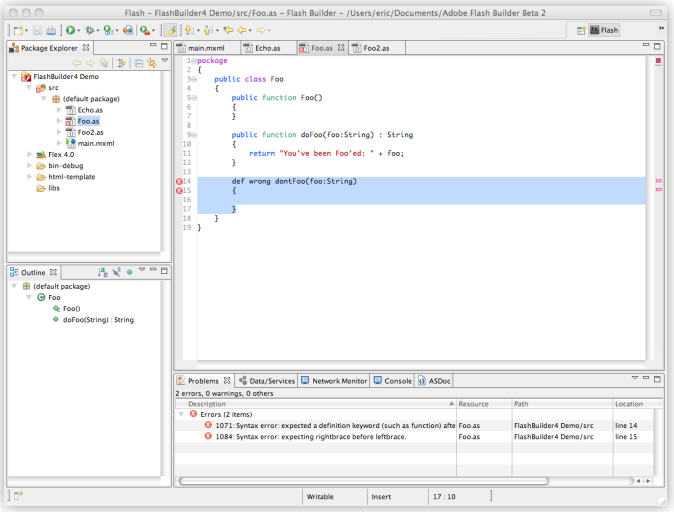
Flex also provides a layout specifically for button bars skins (ButtonBarHorizontalLayout).

You can create your own Layouts by extending one of these layouts, or their base class, LayoutBase.

ActionScript

When you move beyond simple components and need to begin to create object models and complex logic, it is time to use the ActionScript editor. The ActionScript editor is a traditional source code text editor.

The editor includes keyword highlighting, error highlighting, and provides a list of current problems (Errors and Warning) in the view at the bottom of the workspace. The following figure shows a simple ActionScript class with two unresolved problems.



Code Completion

Flash Builder provides code completion for many common scenarios. Code completion can be triggered automatically based on context or manually, using Ctrl-Space. Code completion is automatically triggered after a period or colon, displaying either possible variable and property names, or available types.

Hot Tip

Dynamic Methods

ActionScript is a dynamic language. Therefore, the code completion can only provide guidance for the statically known functions and properties. There will be occasions when code completion will be unable to identify valid functions or properties.

Code Generation

ActionScript is a relatively concise language, avoiding significant amounts of boilerplate code. However, there are still occasions where it is useful to have Flash Builder perform common code generation.

Generate Getter/Setter provides the ability to generate a get and set method for a given variable.

Generate Event Handler generates an event handler method and assigns it to the event property on the MXML component. Generate Service Call generates code to integrated with server classes when using Data Services.

The following table provides a list of where and how these code generation functions can be invoked.

Name	Location	Invoke
Generate Getter/Setter	ActionScript Editor	Right Click (Win)/Ctrl-Click(OSX) on Variable and select Source -> Generate Getter/Setter.
Generate Event Handler	MXML Source View, MXML Design View	Source: Ctrl-Space in event property. Design: Right Click on a Component -> Generate Click Handler OR click the button in the Flex Properties View and select Generate Event Handler.
Generate Service Call	MXML Design View	Right Click on a Component -> Generate Service Call OR click the button in the Flex Properties View and select Generate Service Call.

Refactoring

Flash Builder provides support for some basic refactoring. You can Rename variables and functions, and Rename or Move classes using the menu: Source -> Refactor -> Move

(or Rename). You can also access the Refactor menu from the context menu (Right Click (Win) or Ctrl Click (OS X)). These menu items will open a dialog box allowing you to specify the new name or location.

Hot Tip

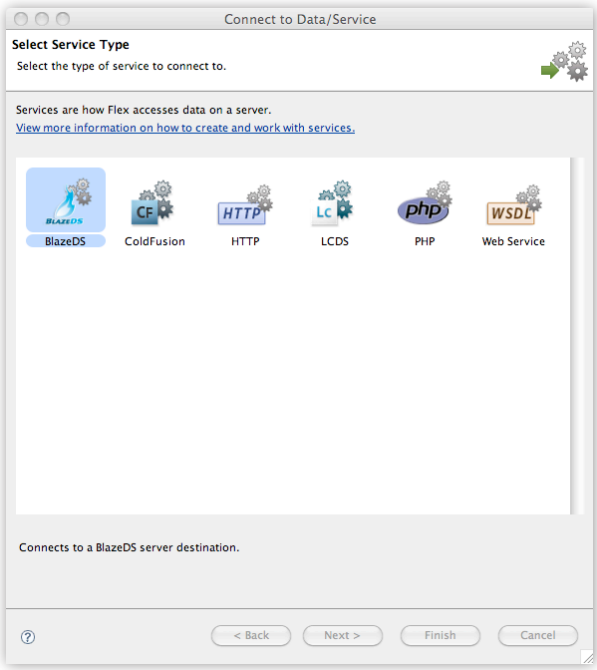
Source Control
Flash Builder supports integration with many different source control management tools. You can install an Eclipse Plug-in for the source control management tool you use.

Data Centric Development

The Flash Builder Data Centric Development feature allows developers to quickly and easily connect Flex applications to a backend service.

To use the Data Centric Development feature, the project must be connected to a service. This can either be done when the project is created, or later when needed.

If a service was not defined, Flash Builder will prompt to create one when the Generate Service Call code generation feature is first used.



Select the type of service you are using. You will then be taken to the Project Properties page where you can setup your service (this the same options and inputs used by the New Project Wizard)

BUILDING

Compiling

Flash Builder performs a background compilation every time a file is saved. This provides real time feedback to compile errors while also encouraging users to save regularly. You can disable this feature by selecting the menu Project -> Build Automatically. You can perform a manual incremental compile at any time using Ctrl-B (Win) or Command-B (OS X).

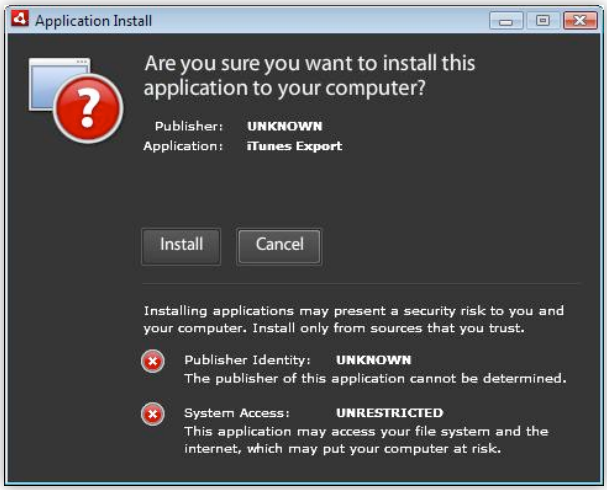
You can perform a clean compile by selecting the Project -> Clean… menu item. You can then opt to rebuild one or more of the projects in your workspace.

Export Release Build

When you are done testing and ready to distribute your application, you can use the Export Release Build command (Project -> Export Release Build…) to create a packaged version of your application.

Web projects are packaged into .swf files that can be uploaded to your web server for distribution. Flash Builder also generates an HTML page and supporting files for your application. You can find the template in the html-template directory. This enables easy deployment of an application, or can serve as a guide for a custom HTML page.

Desktop projects are packaged into an .air file that can be installed on a user's computer. Each Adobe AIR application must be signed using a digital certificate. You can purchase a certificate from a Root Certificate Authority, or generate your own self-signed certificate. If you generate your own certificate, users will receive a warning when they install the application.



Therefore, it is important to use a valid certificate from a Root Certificate Authority when distributing applications to a broad user base.

You can also generate an intermediate file (.airi). This is an unsigned .air file that must be signed (and converted to an .air file) before it can be distributed. This is useful in situations where the person who builds the project is different from the person who is authorized to use the certificate.

RUNNING AND DEBUGGING

Flash Builder provides the ability to launch and debug your application from within Flash Builder. Applications can be launched as a separate process (Run), or Flash Builder can attach to the running application (Debug).

Flash Builder is configured by default to use your primary mxml file as the entry point. However, you can configure multiple Run/Debug configurations using different entry points.

To run the default application, use the menu Run -> Run main (where main is the name of your default mxml file), or use the Ctrl-Shift-F11 (Win) or Command-Shift-F11 (OS X) keyboard shortcut. For web applications this will launch your application in a browser using the html-template. Desktop applications will be run using Adobe's runtime engine.

If you want to debug the application you can launch it using Run -> Debug main, or use the Ctrl-F11 (Win) or Command-F11 (OS X) keyboard shortcut. This will launch the application in the browser or runtime engine, and Flash Builder will connect to the running application, allowing control and introspection.

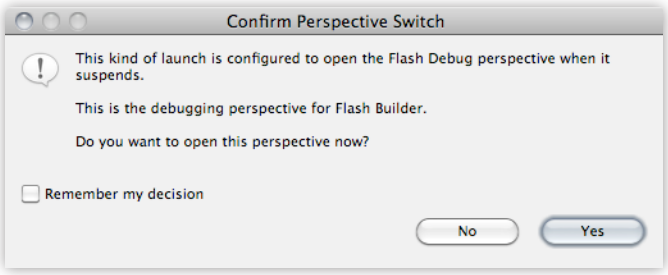
Breakpoints

When debugging an application, you can set breakpoints to indicate you want the application to pause when it reaches a certain line. Flash Builder allows even more control, allowing break points that are conditional on the value of variables or functions.

Hot Tip

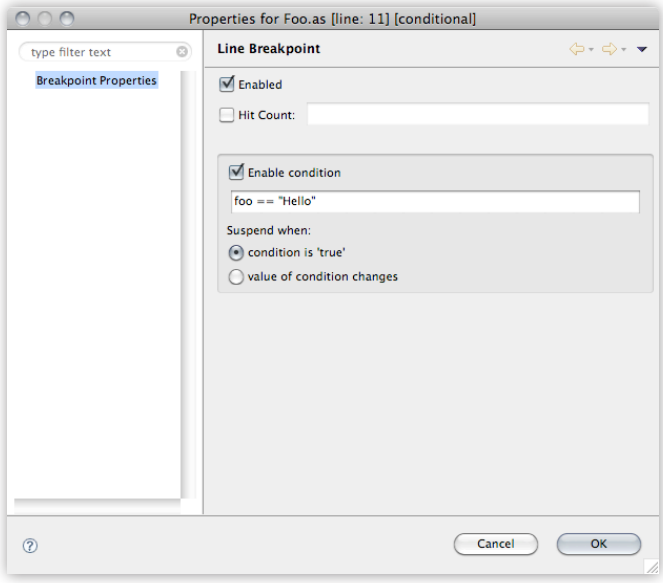
Flex Debug Perspective

Remember to use the Flex Debug perspective when debugging an application. Flash Builder will also prompt you to switch when a breakpoint is hit during debugging.



You can select Remember my decision to automatically switch to the Flex Debug perspective in the future.

Breakpoints can be set by clicking to the left of the line number in the Editor, or using the menu Run -> Toggle Breakpoint or using the keyboard shortcut Ctrl-Shift-B (Win) or Command-Shift-B (OS X). By default the debugger will stop before the code on the selected line is executed. You can make a breakpoint conditional by opening the breakpoint properties (Right-Click on the Breakpoint and select Breakpoint Properties).



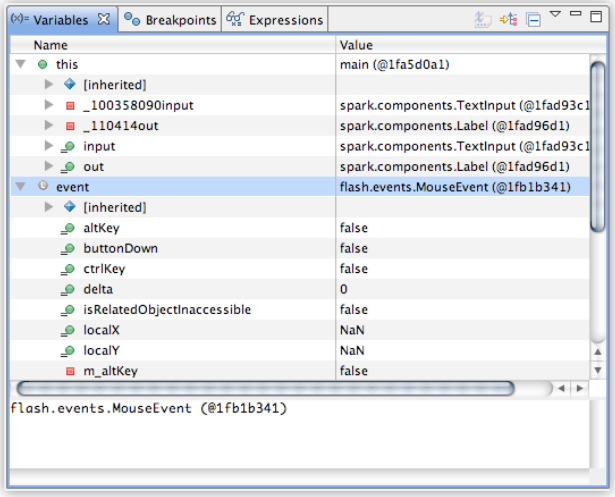
Select the Enabled checkbox and specify the condition that must be true for the breakpoint. You can also choose to specify a value, and the breakpoint will pause when that value changes.

Execution Control

Flash Builder allows fine-grained control over the execution of each line of code while debugging. The Flex Debug perspective provides a toolbar with buttons to step Into, Over, and Out of (Return) the current line. You can tell Flash Builder to run to the current line using the keyboard shortcut Ctrl-R (Win) or Command-R (OS X)

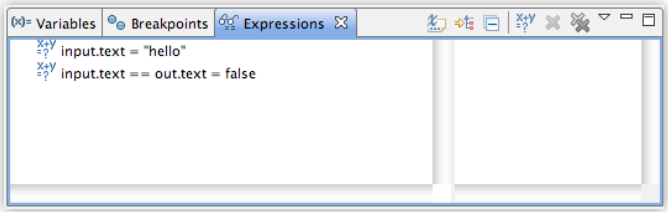
Variables and Expressions

While debugging in the Flex Debug perspective, you can view the current values for your variables using the Variables view.



This view provides a convenient way to explore the variables that exist within the current scope. While this is useful, it can be time consuming to find a specific value during extended debugging sessions or across repeated debugging sessions. The Expressions view allows you to set and view the values of specific expressions (Watched Expressions).

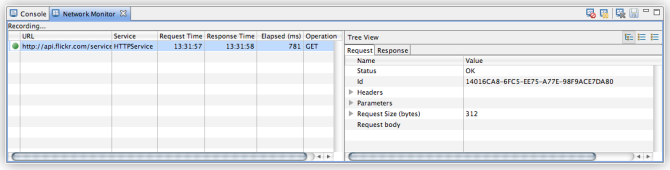
A Watched Expression is a small code expression that produces a single value. Expressions can be simple variables or complex logic including function calls that evaluate to a single value. The following figure shows two watched expressions, the first is a variable while the second is a simple Boolean statement.



Network Monitor

Many applications developed with Flash Builder communicate with external services to read or write data. This can be difficult to debug without additional visibility into what is happening.

Flash Builder includes the Network Monitor view, which provides visibility into the network calls made by the program. The Network Monitor tracks all requests, including the URL, Request and Response Times, Duration, and Operation. For each request, you can inspect the Request and Response using a view similar to the Variables view.



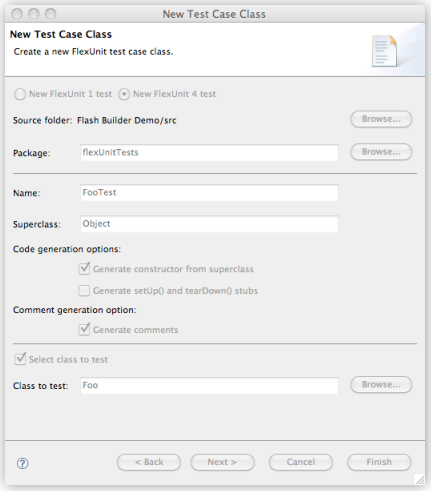
TESTING

Unit Testing is commonly used by development teams to provide automated initial and regression testing. Flex developers can use FlexUnit, an open-source testing framework modeled after the popular JUnit project used for Java Unit Testing. FlashBuilder supports FlexUnit testing by providing integrated code generation and execution features, including the ability to create TestSuite and TestCase classes, and the Run and Debug tests.

Hot Tip

FlexUnit
You can find out more about FlexUnit from the project home page: http://opensource.adobe.com/wiki/display/flexunit/FlexUnit

A FlexUnit test case is an individual method that executes a portion of an application and asserts that the results received were the expected results. Test cases are grouped into a TestCase class. TestCase classes are grouped into a TestSuite class, allowing groups of test to be organized and run as separate suites. An individual TestCase class could be added to multiple TestSuites.



FlashBuilder provides the ability to create TestSuite and TestCase classes from the menu File -> New -> Test Suite Class (or Test Case Class). These menu items open a Dialog box that allows you to specify the options to create the class.

FlashBuilder supports the creation of both FlexUnit 1 and FlexUnit 4 code. After selecting the appropriate package, class name, and other parameters, you can select a specific class to test, in this case Foo. The next step allows you to select individual methods to test.

FlexUnit tests can be run from the menu Run -> Run -> FlexUnit Tests.

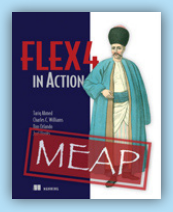
Try Flash Builder 4: www.adobe.com/go/try_flashbuilder

ABOUT THE AUTHOR

Eric Daugherty is a Senior Architect at Gorilla Logic, Inc. (www.gorillalogic.com) where he develops enterprise applications using Flex and Java. He enjoys solving problems and is passionate about learning new technologies and tools. You can read about his open source projects and blog at:

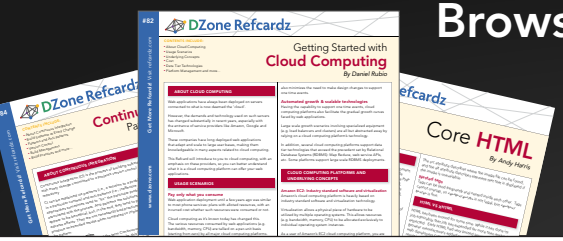
http://www.ericdaugherty.com/blog/

RECOMMENDED BOOK



Flex 4 in Action is an easy-to-follow, hands-on Flex 4 tutorial. Revised and updated from the previous edition on Flex 3, this book is chock-full of examples, goes beyond feature coverage, and helps readers put Flex to work in real day-to-day tasks.

BUY NOW
books.dzone.com/books/flex4



Browse our collection of over 100 Free Cheat Sheets

Free PDF

Upcoming Refcardz
Network Security
Hadoop
UML
Subversion

DZone

DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. **"DZone is a developer's dream,"** says PC Magazine.

DZone, Inc.
140 Preston Executive Dr.
Suite 100
Cary, NC 27513

888.678.0399
919.678.0300

Refcardz Feedback Welcome
refcardz@dzone.com

Sponsorship Opportunities
sales@dzone.com

ISBN-13: 978-1-934238-71-4
ISBN-10: 1-934238-71-6
50795
9 781934 238714
$7.95