DZone Refcardz

brought to you by...
Adobe

CONTENTS INCLUDE:

Adobe Flash Builder 4.5:

Develop for the Web, Desktop and Mobile

By Holly Schinsky

ABOUT FLASH BUILDER 4.5

Adobe® Flash® Builder 4.5®, formerly Adobe® Flex® Builder®, is an Integrated Development Environment (IDE) from Adobe Systems Inc. for ActionScript® and Flex development. It's built on top of Eclipse™, an open-source, extensible development platform and a popular Java IDE. Because of this, Flash Builder 4.5 inherits an impressive list of capabilities and is a familiar tool for many developers. The latest version now includes support for building mobile applications as well as many developer productivity features.

> **Hot Tip**
>
> Adobe® Flash® Builder® 4.5 is an Eclipse-based development tool for rapidly building expressive mobile, web, and desktop applications using ActionScript and the open-source Flex framework.

This Refcard outlines how to use Flash Builder 4.5 to develop Flex applications, including creating new projects for mobile and other platforms as well as developing, building (compiling), testing, and debugging Flex applications.

Getting Adobe Flash Builder 4.5

Go to the Adobe download site (http://www.adobe.com/products/flash-builder.html) and choose the appropriate download file for your operating system.

The downloaded file is an installer. For Windows, simply run the downloaded file (.exe). For Mac 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 4.5 can be run from the installed location (Program Files on Windows or Applications on OS X) or by launching the Eclipse plug-in version. There is also a new Flash Builder 4.5 for PHP version available for PHP developers.

When Adobe Flash Builder 4.5 is first launched, it prompts you for a serial number. If you don't have one, 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**
>
> **Build Mobile applications with Flash Builder 4.5**
> You can now build projects for mobile devices using Flex or ActionScript with Flash Builder. When creating a new project, you can choose to create a new Flex Mobile project or a new ActionScript Mobile project.

CREATE A WEB/DESKTOP PROJECT

To create a new project targeting the web or desktop, choose **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.

Application Type

Flash Builder supports two applications types from this wizard: Web and Desktop. Web projects create SWF files that can be distributed via a web server and run with Adobe Flash Player in a client browser. Desktop projects create AIR files that can be downloaded and installed as standalone 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.

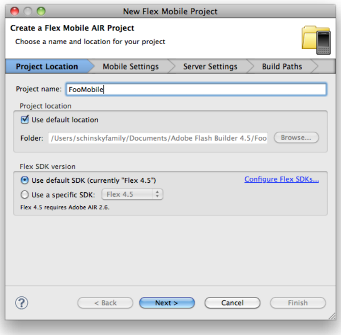
Server Technology

Flash Builder 4.5 supports integration with several server technologies, including Java, PHP, ASP.Net and ColdFusion. Selecting one of these options enables Flash Builder to provide automatic configuration for remote calls. If an option other than None/Other is selected, additional information will be collected in subsequent steps in the wizard. The specific data needed is dependent on the selected technology.

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

CREATE A MOBILE PROJECT

To create a new project targeting mobile devices, choose **File > New > Flex Mobile Project** option. The first step in the wizard (Project Location) is similar for the first step in creating a regular Flex Project. Type a name for the project and click Next to choose your mobile project settings.



New Flex Mobile Project

Create a Flex Mobile AIR Project
Choose a name and location for your project

Project name: FooMobile

Project location
☑ Use default location
Folder: /Users/schinskyfamily/Documents/Adobe Flash Builder 4.5/Foo

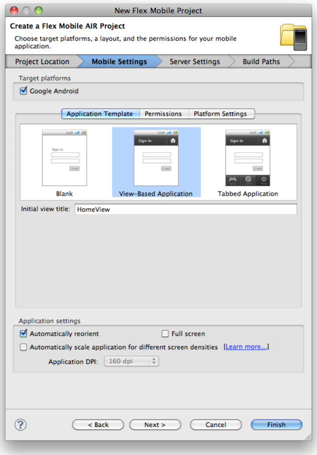
Flex SDK version
● Use default SDK (currently "Flex 4.5")
○ Use a specific SDK: Flex 4.5
Flex 4.5 requires Adobe AIR 2.6.

Adobe
Adobe® Flash® Builder® 4.5
Deliver expressive, easy-to-use applications on the leading mobile and web platform and devices
Fx
FB
ADOBE® FLASH® PLATFORM
Download your free trial today:
www.adobe.com/go/flashbuilder45_try

Mobile Settings

The mobile settings step in the wizard contains three tabs: Application Template, Permissions, and Platform Settings.

First choose the type of application to create; it will usually be a View-Based application or a Tabbed Application. Then type a title for the initial view to be shown or use the default. This title is shown in the title bar content of the application at runtime unless otherwise changed.



Create a Flex Mobile AIR Project

Choose target platforms, a layout, and the permissions for your mobile application.

Target platforms

☑ Google Android

Initial view title: HomeView

Application settings

☑ Automatically reorient ☐ Full screen

☐ Automatically scale application for different screen densities [Learn more...]

Application DPI: 160 dpi

Hot Tip: Selecting the Mobile Application Type

The Tabbed Application option should be selected for more complex applications that might need several stacks of views, through which each set of views (via the tabs) can be navigated separately.

Next select if you want the application to automatically reorient (change from portrait to landscape when turned), if it should use the full screen of the device, and if you want it to scale for different screen densities automatically.

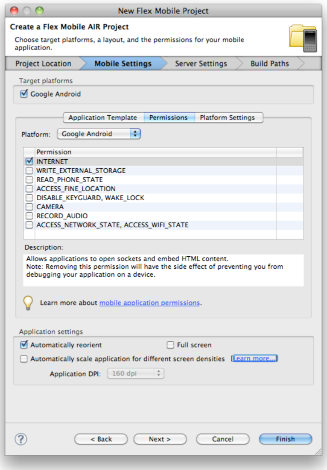
Hot Tip: Automatic Application Scaling and DPI

When you enable automatic scaling, Flex provides optimal viewing capabilities across different screen densities. For example, if you specify the target DPI value as 160 and enable automatic scaling, when you run the application on a device with a DPI value of 320, Flex automatically scales the application by a factor of 2.

Select the Permissions tab to set any permissions needed by the application. These are typically specific to the Android OS. Permissions are needed to access certain device hardware and capabilities. Click on a permission to see its description.

When you're finished setting permissions click the Platform Settings button.

Use the Platform Settings tab to set any specific settings for your target platform. For instance, if you are developing for iOS (when available), you can choose from a target device of iPhone/iTouch or iPad.



Create a Flex Mobile AIR Project

Choose target platforms, a layout, and the permissions for your mobile application.

Target platforms

☑ Google Android

Platform: Google Android

	Permission
☑	INTERNET
☐	WRITE_EXTERNAL_STORAGE
☐	READ_PHONE_STATE
☐	ACCESS_FINE_LOCATION
☐	DISABLE_KEYGUARD, WAKE_LOCK
☐	CAMERA
☐	RECORD_AUDIO
☐	ACCESS_NETWORK_STATE, ACCESS_WIFI_STATE

Description:

Allows applications to open sockets and embed HTML content.
Note: Removing this permission will have the side effect of preventing you from debugging your application on a device.

Learn more about mobile application permissions.

Application settings

☑ Automatically reorient ☐ Full screen

☐ Automatically scale application for different screen densities [Learn more...]

Application DPI: 160 dpi

Once all mobile settings have been specified, click Next to move to the Server Settings screen.

Server Settings

This screen is used when you need to configure server-side options for your application (accessing a Java, ColdFusion, or PHP remote service, for instance). Type any additional server-side configuration settings here if needed and click Next.

Build Paths

The last screen in the wizard is for Build Path settings. These are typically left as their default values. When you click Finish, Flash Builder will create your project and open it for editing.

DEVELOP

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

Workbench

The workbench comprises 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, and Problems View among 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 4.5 is built on Eclipse and inherits many of its features and shortcuts. For more information on Eclipse, see the *Getting Started with Eclipse Refcard*. http://refcardz.dzone.com/refcardz/getting-started-eclipse

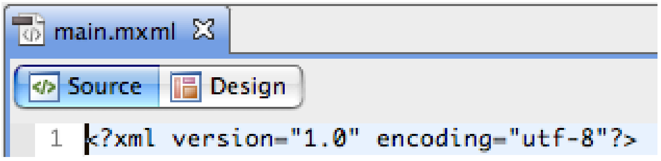
Building Flex Applications

Flex applications consist of two different types of source files. MXML files (.mxml) contain the markup used to arrange visual components as well as non-visual components and ActionScript code. ActionScript files (.as) contain ActionScript code (classes, interfaces, and so on). You can achieve the same functionality with either type of file, but they are tailored to different use cases.

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

MXML

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



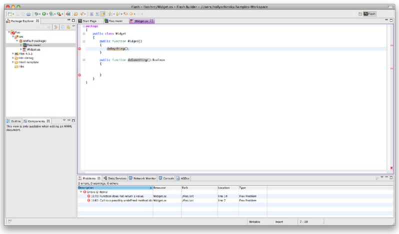
```
<?xml version="1.0" encoding="utf-8"?>
```

Source mode provides an XML editor with code completion, syntax highlighting, and error highlighting.

Hot Tip: ASDoc Tool Tips

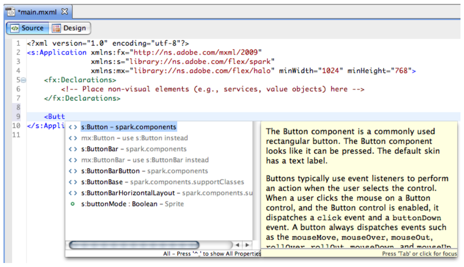
You can quickly view the ActionScript documentation as a ToolTip in the MXML Source Editor and ActionScript Editor by hovering the over a Flex SDK class or method

ActionScript



When you move beyond simple components and need 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, which includes keyword highlighting and error highlighting. It also provides a list of current problems (Errors and Warnings) in the view at the bottom of the workspace.

Design Mode



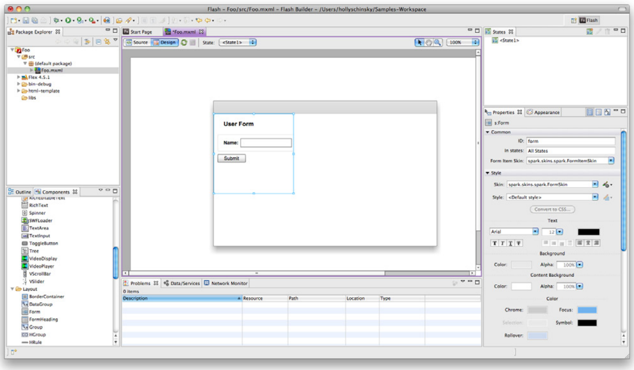
```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/halo" minWidth="1024" minHeight="768">
    <fx:Declarations>
        <!-- Place non-visual elements (e.g., services, value objects) here -->
    </fx:Declarations>

    <Butt
</s:Appli
```

s:Button – spark.components
mx:Button – use s:Button instead
s:ButtonBar – spark.components
mx:ButtonBar – use s:ButtonBar instead
s:ButtonBarButton – spark.components
s:ButtonBase – spark.components.supportClasses
s:ButtonBarHorizontalLayout – spark.components.su
s:buttonMode : Boolean – Sprite
All – Press '^.' to show All Properties
Press 'Tab' or click for focus

The Button component is a commonly used rectangular button. The Button component looks like it can be pressed. The default skin has a text label.

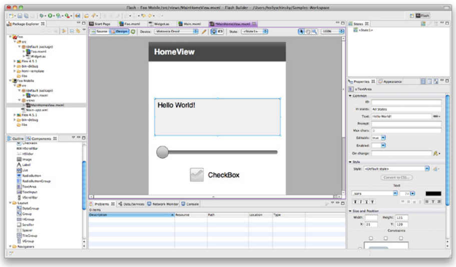
Buttons typically use event listeners to perform an action when the user selects the control. When a user clicks the mouse on a Button control, and the Button control is enabled, it dispatches a click event and a buttonDown event. A button always dispatches events such as the mouseMove, mouseOver, mouseOut, rollOver, rollOut, mouseDown, and mouseUp

Design mode provides a visual preview of your application. It's very useful for laying out 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 Components View at the bottom left. You can drag and drop these onto your application and then adjust their size and position in the layout. You can also change between different UI states easily by selecting a different state in the drop-down list at the top.



Hot Tip

Mobile Application Design Mode
You can use Design Mode when building mobile applications. You can even select a target device by using the drop-down list at the top.



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 absolute 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, you cannot (usefully) resize the application. It can also be very difficult to internationalize such 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 bar skins (ButtonBarHorizontalLayout).

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

PRODUCTIVITY FEATURES

Flash Builder 4.5 includes numerous coding productivity features, including several that were not available in earlier versions.

Code Completion

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

Hot Tip

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

Code Templates

Code templates allow you to quickly insert snippets of code commonly used in ActionScript, MXML, or CSS such as a for loop, script block, or Web Service call. A number of predefined templates are included out of the box and you can add or edit them as desired by choosing Preferences > Flash Builder > Editors > Code Templates > ActionScript/CSS/MXML.

Quick Fix/Assist

The Edit menu (or right-click menu) contains a new option for performing a Quick Fix/Assist to help fix or update your code based on the current context. Quick Assist (press Ctrl +1) can be used for the following cases:

- Split variable declaration
- Assign to variable
- Convert local variable to field
- Generate Getter/Setter
- Rename in workspace
- Rename in file
- Organize imports

Metadata code completion

Metadata code completion provides content assist help when the open square bracket '[' character is typed.

Override/Implement Methods

The Source menu has a new option to Override/Implement Methods that allows you to quickly choose methods you want to implement or customize.

Code Generation/Generate from Usage

ActionScript is a relatively concise language, which enables you to avoid significant amounts of boilerplate code. However, there are still occasions where it is useful to have Flash Builder generate commonly used code.

Generate Getter/Setter generates a get and set method for a given variable.

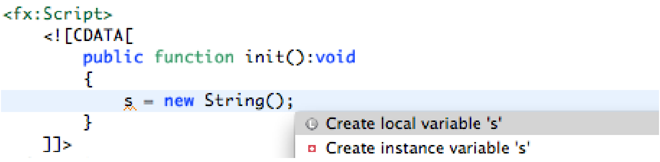
Generate Event Handler generates an event handler method and assigns it to the event property on the MXML component.

Generate Service Call generates code to integrate with server classes when using Data Services.

The following table shows how these generators can be invoked.

Name	Location	Invoke
Generate Getter/ Setter	ActionScript Editor	Right-click (Win)/Ctrl-click(OS X) the 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 the Component and select Generate Click Handler OR click the button in the Flex Properties View and select Generate Event Handler
Generate Service Call	Relative	Right-click the component and select Generate Service Call OR click the button in the Flex Properties View and select Generate Service Call

You can also use the Quick Assist popup menu (Ctrl +1) to create a variable, method, class, or interface based on context.



```
<fx:Script>
    <![CDATA[
        public function init():void
        {
            s = new String();
        }
    ]]>
```

> **Hot Tip**
> **Other Coding Enhancements**
> For detailed information on coding productivity features in Flash Builder 4.5, see Adobe's article on Devnet.

Refactoring

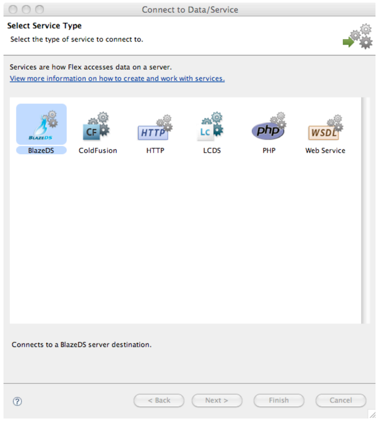
Flash Builder provides support for some basic refactoring. You can rename variables and functions and Rename or Move classes by choosing **Source > Refactor > Move** (or **Rename**). You can also access the Refactor menu from the context menu.

> **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 enables developers to quickly and easily connect Flex applications to a back-end service.

To use data-centric development, the project must be connected to a service. This can either be done when the project is created or later when needed.



Connect to Data/Service

Select Service Type

Select the type of service to connect to.

Services are how Flex accesses data on a server.

View more information on how to create and work with services.

Connects to a BlazeDS server destination.

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

Select the type of service you are using. You will then be able to set up your service using the same options and settings available in the New Project wizard.

BUILD

Compiling

By default, Flash Builder performs a background compilation every time a file is saved. This provides real-time feedback on compile errors while also encouraging users to save regularly. You can toggle this feature by choosing Project - Build Automatically. You can perform a manual incremental compile at any time by pressing Ctrl-B (Win) or Command-B (OS X).

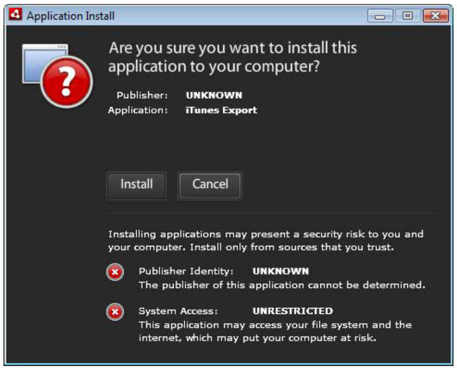
To perform a clean compile, choose **Project > Clean**. You can then select the project or projects in your workspace to build.

Export Release Build

When you are done testing and ready to distribute your application, use the Export Release Build command (choose **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 template enables easy deployment of an application, or it can serve as a guide for a custom HTML page.

Desktop projects are packaged into AIR files that can be installed on a user's computer. Each 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.



Application Install

Are you sure you want to install this application to your computer?

Publisher: UNKNOWN
Application: iTunes Export

Installing applications may present a security risk to you and your computer. Install only from sources that you trust.

Publisher Identity: UNKNOWN
The publisher of this application cannot be determined.

System Access: UNRESTRICTED
This application may access your file system and the internet, which may put your computer at risk.

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 when the person who builds the project is different from the person who is authorized to use the certificate.

RUN AND DEBUG

You can 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 set up multiple Run/Debug configurations using different entry points.

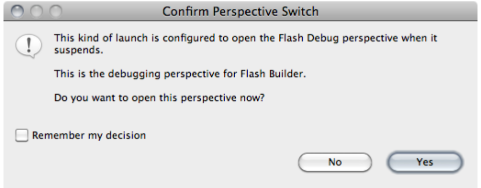
To run the default application, choose **Run > Run main** (where main is the name of your default MXML file). For web applications this will launch your application in a browser using the html-template. Desktop applications will be run using the AIR runtime.

If you want to debug the application you can launch it by choosing **Run > Debug main**. Or you can 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, enabling control and introspection for debugging.

Breakpoints

When debugging an application, you can set breakpoints to indicate you want the application to pause when it reaches a certain line of code. Flash Builder supports breakpoints 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


Confirm Perspective Switch

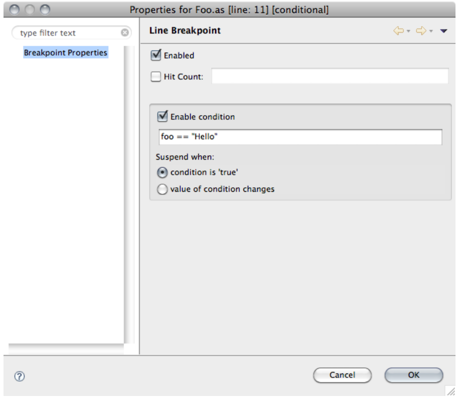
This kind of launch is configured to open the Flash Debug perspective when it suspends.

This is the debugging perspective for Flash Builder.

Do you want to open this perspective now?

□ Remember my decision

Breakpoints can be set by clicking to the left of the line number in the editor, choosing **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 editing the breakpoint properties (right-clicking the Breakpoint and selecting Breakpoint Properties).


Properties for Foo.as [line: 11] [conditional]

Line Breakpoint

☑ Enabled
□ Hit Count:

☑ Enable condition
foo == "Hello"

Suspend when:
● condition is 'true'
○ value of condition changes

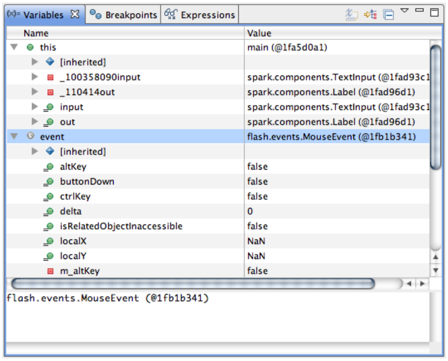
Select the Enabled checkbox and specify the condition that must be true for the breakpoint. You can also choose to specify a value, and the breakpoint will pause when that value changes.

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 have Flash Builder 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.

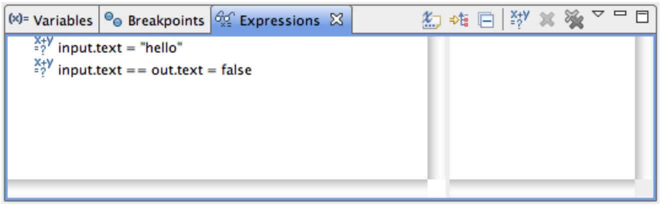


Name	Value
this	main (@1fa5d0a1)
[inherited]	
_100358090input	spark.components.TextInput (@1fad93c1
_110414out	spark.components.Label (@1fad96d1)
input	spark.components.TextInput (@1fad93c1
out	spark.components.Label (@1fad96d1)
event	flash.events.MouseEvent (@1fb1b341)
[inherited]	
altKey	false
buttonDown	false
ctrlKey	false
delta	0
isRelatedObjectInaccessible	false
localX	NaN
localY	NaN
m_altKey	false

```
flash.events.MouseEvent (@1fb1b341)
```

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 enables you to set and view the values of specific expressions, known as 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.

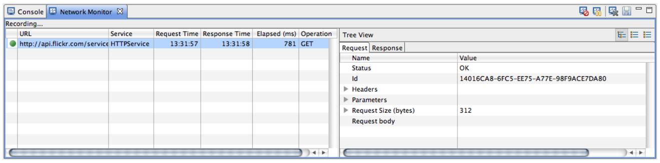


```
input.text = "hello"
input.text == out.text = false
```

Network Monitor

Many applications developed with Flash Builder communicate with external services to read or write data. This communication 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 application under development. 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.



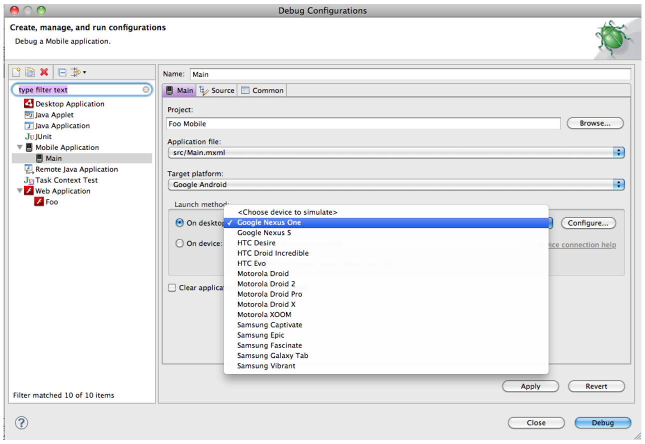
Recording...

URL	Service	Request Time	Response Time	Elapsed (ms)	Operation
http://api.flickr.com/service	HTTPService	13:31:57	13:31:58	781	GET

Name	Value
Status	OK
Id	14016CA8-6FC5-EE75-A77E-98F9ACE7DA80
Headers	
Parameters	
Request Size (bytes)	312
Request body	

RUN AND DEBUG FOR MOBILE

Flash Builder 4.5 can package, deploy, and launch your mobile application by using the run or debug configuration wizards. You can choose to preview and debug it in the included Adobe AIR runtime emulator or directly on your locally connected mobile device over USB (for Android devices). You can use all of the traditional debug functionality, including breakpoints, as you would on the desktop; however profiling is not supported for mobile. In the run/debug configuration dialog box you can choose a target platform and device as shown below.


Debug Configurations

Create, manage, and run configurations
Debug a Mobile application.

Name: Main

Project: Foo Mobile

Application file: src/Main.mxml

Target platform: Google Android

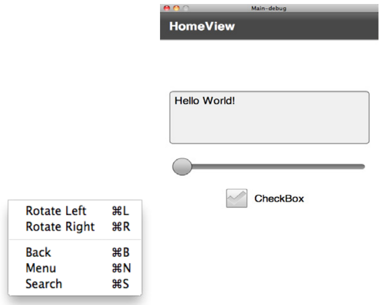
Launch method:
● On desktop: Google Nexus One
○ On device:

□ Clear applica

Filter matched 10 of 10 items

Emulator

When your application is run on the emulator you can simulate a rotation, menu, search, or home button press via the (AIR debug launcher) menu.



Packaging applications for deployment

To package a mobile app, use the Export Release Build process just as you would to prepare desktop and web applications for deployment; however, when you export a release build of a mobile project, Flash Builder packages the build as a native installer, not as an AIR file. For Android, Flash Builder produces an .apk file that looks the same as a native Android application package. On Android, users who do not already have the AIR runtime installed will be prompted to download and install it the first time they run any AIR application. This step is not necessary on iOS, where the runtime is bundled into the packaged application; or on BlackBerry Tablet OS, where the runtime is built into the operating system.

TEST

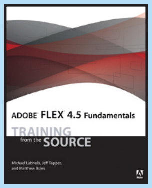
Unit Testing is a commonly used by development teams to provide automated initial and regression testing. Flex developers can use FlexUnit, an open-source testing framework modeled after the popular JUnit project used for Java Unit Testing. Flash Builder supports FlexUnit testing by providing integrated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

Visit http://flex.org for more information, tutorials, and articles.

ABOUT THE AUTHOR

Holly Schinsky is a senior software engineer with 14 years of experience in development, including Flex/AIR/ActionScript, Java and Java EE as her primary focus. She was one of the authors of the Tour de Flex AIR and web apps, and wrote the Tour de Flex Eclipse plug-in for Adobe. She has worked on various Adobe projects over the years, including an XForms to XFA conversion tool for the Adobe LiveCycle Designer application and the Adobe AIR Launchpad application available on Adobe Labs. She has an extensive background in server-side programming, including being part of the small development team that wrote the workflow engine originally behind LiveCycle ES. In recent years, Holly has become heavily involved in the Flex community, including working on the Attest Flex/AIR Certification study AIR application in addition to continuing to support Tour de Flex and AIR Launchpad.

RECOMMENDED BOOK



Adobe Flex: Training from the Source is the best-selling and most trusted resource for learning about Adobe Flex. Written by a team of authors with practical experience as consultants, mentors and developers of courseware, this book/CD uses project-based tutorials, and is designed to teach beginning Flex developers the details of building and architecting real-world rich internet applications using Flash Builder incorporating MXML and ActionScript 3.0. The book includes a CD that contains all the files used in the lessons, plus completed projects for comparison. This latest edition includes complete coverage of new Flex 4.5 features, such as new enhancements to the Spark architecture and component set. It will also show you how to take advantage of the improvements to core Flex infrastructure for large application development.

Pre-order Now!

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, cheat sheets, 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: 978-1-936502-40-0
ISBN-10: 1-936502-40-2
50795
9 781936 502400

$7.95

Version 1.0