CONTENTS INCLUDE:

- What is Flex?
- Building Applications with Adobe® Flash® Builder™ 4 and PHP
- Connecting to Web Services
- Developing your Application
- Hot Tips and more...

Getting Started with Integrating PHP and Flex

By Marco Tabini

WHAT IS FLEX?

Flex is an open-source framework developed and distributed by Adobe Systems. It is based on the Adobe® Flash Platform and primarily provides a streamlined approach to the development of Rich Internet Applications.

Flex eliminates many of the designer-oriented features of Flash in favor of establishing a development environment that caters more to programmers. As such, you will find that Flex encompasses many of the concepts that you are already familiar with if you have developed front-end systems using JavaScript or, indeed, most other GUI programming environment, allowing you to take advantage of the underlying Flash infrastructure without having to worry about concepts like timelines, assets, and so on.

Flex is multi-platform—this means that, with some exceptions, you can run a Flex application on any platform that supports Adobe Flash Player. If your users run on Windows, OS X or Linux and their browsers have a recent version of the Flash Player plug-in installed, they will also be able to run your Flex applications without a problem.

Because Flex is open source, there is no cost associated with creating and distribution applications that are based on it.

Hot Tip You can download Adobe Flex SDK for free directly from the Adobe website at http://www.adobe.com/products/flex/

What is Adobe AIR?

The Adobe Integrated Runtime (Adobe AIR) is a companion technology to the Flex framework that extends the functionality provided by the latter into desktop application development. With AIR, you can build Flex applications that can be deployed as native applications on your user's machines, thus gaining all the advantages of running in a desktop environment.

Like Flex, AIR is also cross-platform, which means that you can write your code once and immediately deploy it across multiple operating systems. Because they run natively rather than in a web browser, AIR applications also gain access to functionality that is usually restricted by the Flash Player's security model, such as local file manipulation, unrestricted access to the network, and so forth.

What is Adobe Flash Builder?

Flash Builder 4 is Adobe's IDE for developing Flex and AIR applications. Although Flash Builder 4 is not required in order to compile or run a Flex-based application, it significantly simplifies the process of Flex development by providing an integrated environment that includes code intelligence, real-time analysis, compilation support, live debugging and much more.

Flash Builder 4 is based on the open-source Eclipse IDE and can either be downloaded as a standalone product or as a plug-in for the latter. Like Eclipse, Flash Builder 4 is also cross-platform and runs on both Windows and OS X.

Hot Tip You can download a 60-day trial of Flash Builder 4 from the Adobe website at http://www.adobe.com/products/flex/

BUILDING APPLICATIONS WITH FLASH® BUILDER™ 4 AND PHP

Even though Flex is based on Flash, you don't need to be proficient in the latter in order to use the former.

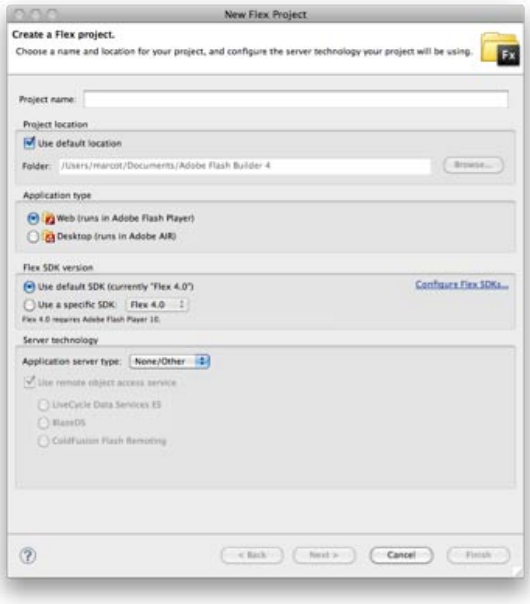
Flex uses a language called ActionScript3 (AS3), which is itself derived from the ECMAScript standard. ECMAScript is the same basic definition on which JavaScript is based—therefore, if you have any familiarity with browser programming, it's likely that you will find your bearings in AS3 very quickly.

Flex applications are based on the concept of component. A component defines a container of behaviors and, optionally, of a user interface representation. Components can be visual or non-visual, depending on whether the provide an interface of some kind (like a button) or just functionality of some kind (like a library for connecting to a remote server).

The visual structure of a component can be easily defined using MXML, Adobe's specialized brand of XML. Other than the use of specific namespaces, MXML is nothing more than well-formed XML code; by nesting multiple components, you can create complex GUIs without ever writing a line of code.

Creating Flex Applications

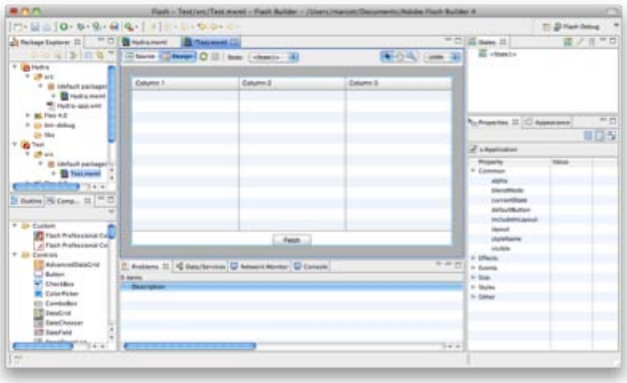
Flash Builder 4 makes creating new applications as easy as following the steps of its New Application Wizard. Simple select New ® Flex Project from the File menu, then choose a name and type for your application. If you intend to write code that will be executed inside a browser, choose "Web" for your application type; if, on the other hand, you want to build a desktop application, choose "Desktop" instead.



You newly-created Flex project will contain a component that represents the application's main entry point:

```
<?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/mx">

  <fx:Declarations>
  </fx:Declarations>
</s:Application>
```

From here on, you can add more components to your application simply by typing the MXML code that represents them in the appropriate order. However, Flash Builder 4's strength resides in its visual layout editor, which allows you to arrange and wire the different components that make up your user interface using a WYSIWYG approach. For example, you can add a DataGrid object to show the contents of a data structure and then a button to fetch the data–all enclosed in a VGroup object to provide the overall layout:



Flex's powerful layout capabilities, like the rest of the framework, are also designed to be developer-friendly and are almost entirely based on standard CSS, with a few exceptions designed to make automated positioning easier.

Useful Flex Components

Flex provides a rich ecosystem of components that can be easily expanded to meet your needs. While many of its components are designed to closely mimic their HTML cousins, there are also a number that provide unique functionality. For example:

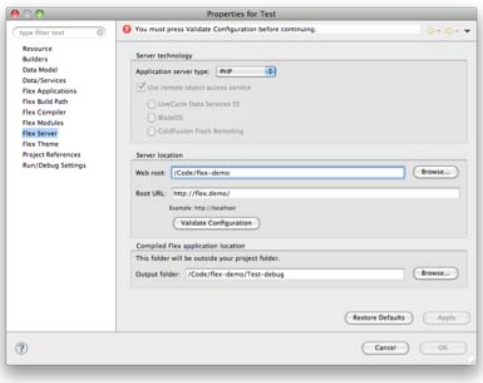
- **AdvancedDataGrid** and **DataGrid**: display tabular data in a rich, editable environment.
- **DateChooser** and **DateField**: simplify the process of choosing and formatting date values.
- **NumericStepper**: allows the user to increment and decrement a numeric value.
- **RichEditableText**: allows you to create a rich text editor with support for several formatting options (e.g.: italic, bold, etc.).
- **VideoDisplay/VideoPlayer**: provide fully-featured and completely skinnable video players that can be embedded directly in your application.
- **HSlider/VSliders**:allows the user to select a value by sliding a selector across a horizontal or vertical range.
- **HGroup/VGroup**: automatically arrange their content horizontally or vertically. Combine to easily create complex layouts.
- **Accordion/TabNavigator**: allow multiple content panels to stack and be displayed one at a time.
- **MenuBar/TabBar**: add a menu or tab bar to your component.
- **Flex Charting**: adds powerful charting abilities to your applications.

Hot Tip While Flex Charting and AdvancedDataGrid were only available in the Professional version of Flex Builder 3, they are now included in all versions of Flash Builder 4.

Connecting to PHP with AMF Introspection

One of the most interesting features of Flex is that it is designed to provide powerful data connectivity capabilities in a variety of formats and protocols, including XML, SOAP, JSON and Adobe's own AMF. In fact, in most cases you will be able to use Flash Builder 4 to connect your Flex application to a PHP backend without having to write a single line of code in either!

In order to use Flash Builder 4's PHP-aware functionality through AMF, you need to have direct access to the root directory of a copy of your site—either through a network share or on your local computer. To get started, select Connect to Data/Service… from the Data menu, then choose PHP from the list of available connectors. Flash Builder 4 will ask you to confirm that your project type is set to PHP, then specify both the location of your site's server root and its URL:



At this point, you can click "Validate Configuration" to make Flash Builder 4 run a simple test to determine whether it can access your site as expected, then OK to complete the set up process.

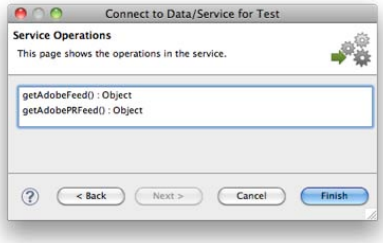
Now, Flash Builder 4 will ask you to choose a PHP class that will provide the entry point to which your Flex application will connect. If your code is not encapsulated in classes, you could create some simple stubs for the purpose of providing services to your Flash Builder 4 code. The wizard will automatically fill in the defaults for you based on the name of the PHP file that you select.

Hot Tip In order for this system to work, it is important that the class you want to import be stored in a file of the same name [e.g.: Index inside index.php]. Otherwise, Zend_AMF will be unable to find it.

At this point, if you website does not include a copy of Zend_AMF, a Zend Framework module that Flash Builder 4 uses to marshal data, remote procedure calls and error management, you will be asked to download and install a copy. This is required because Flash Builder 4 makes use of Action Message Format (AMF), a protocol that PHP does not support by default.

Hot Tip Your application does not need to use Zend Framework in order to take advantage of AMF—Flash Builder 4 will only use Zend_AMF in order to communicate with your server, independently of the rest of your code.

Flash Builder 4 will introspect your code and discover which methods the service makes available:



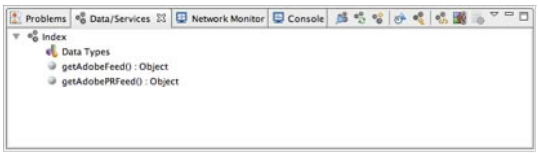
Once you click Finish, the wizard will create a series of classes inside your project that provide all the necessary functionality required to connect to your project and execute your web service.

Hot Tip Remember that it is your responsibility to provide a security layer where required—for example, by passing a secure key to the service as appropriate.

Establishing Data Connections

Your application is now capable of connecting to the PHP backend, but the data that the latter provides is not yet wired to any of the controls.

Luckily, this, too, is something that can be done without writing a line of code. You can, instead, use the Data/Services panel, visible in the bottom tray of the Flash Builder 4 window, where all the remote data services defined in your application are visible:



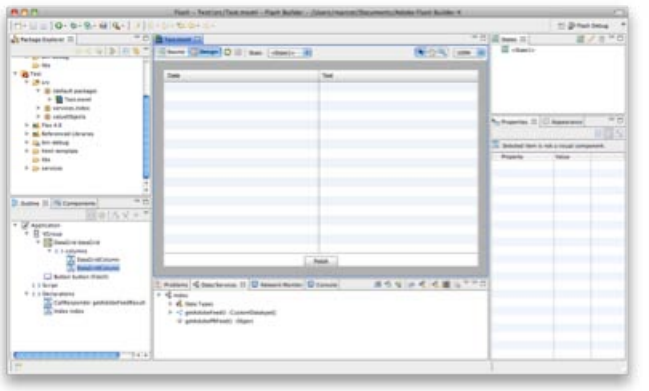
All you need to do in order to connect the data returned by a service call to any of your components is to simply drag it from the Data/Services panel to the component. Flash Builder 4 will ask you whether you intend to create a new call, or use an existing one. In the former case, you will first need to specify the type of data returned by the remote service call, because the data connection wizard has no way of determining it by means of static analysis of your code.

Flash Builder 4 can, however, auto-detect the data type returned by a service call by making a call to it, or you can specify it by hand. Where a sample service call will have no adverse effects on your PHP backend, allowing the wizard to determine the return type automatically is usually sufficient and produces all the infrastructure required to handle the data on the Flex side.

Hot Tip

In order for your data to be used in a Flex application, it must conform to all the appropriate AS3 rules—for example, you cannot return objects with properties whose names are reserved AS3 keywords like protected or private, even if those are perfectly acceptable in PHP.

Your Flex application now has access to all the data returned by your service. If, for example, you drag a service on to a DataGrid component, the latter will be automatically populated with all the appropriate data columns—all you need to do is remove those you don't want displayed and rename the header of the others to the proper human-readable format:



Your application is now fully functional—if you execute it, you will see that the data service is automatically called as soon as the DataGrid object finishes loading. If the call is successful, the data is immediately loaded and displayed.

If you prefer to add a manual method of refreshing the information, you can simply drag the appropriate data call on to the button—this will create all the code needed so that, when the user clicks on it at runtime, the service will be called again and all the data automatically updated.

Differences between PHP and AS3 Data

While much of the data types are interchangeable between AS3 and PHP, there are some notable differences.

- **Integer values** in PHP can either be 32- or 64-bit long, whereas, in AS3, they are always 64 bits. Therefore, you must be prepared for the fact that a numeric value passed from AS3 to PHP may be represented as a float even if it is, in fact, just a large integer.
- **String values** in AS3 are always Unicode-compliant. It is up to you to ensure Unicode compliance on the PHP side.
- **Array values** in AS3 can only have contiguous numeric keys starting at zero. If your PHP arrays have string keys or non-contiguous numeric keys, they will be represented as objects in AS3.

  You should avoid passing objects into AS3 that contain members whose keys are reserved keywords, as handling them will be inconvenient—and many of Flash Builder 4's facilities will refuse to work with them.

Under most circumstances, these issues are unlikely to affect your application because both AS3 and PHP have a significant amount of flexibility

CONNECTING TO WEB SERVICES

SOAP

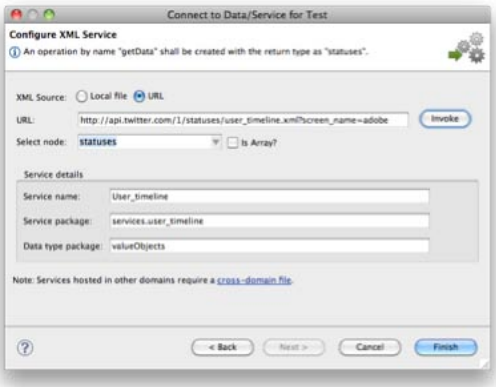
AMF is not your only choice when it comes to external connectivity from Flash Builder 4—almost exactly the same functionality can just as easily be used to connect to an XML web service powered by SOAP.

You can start the process by selecting Connect To Data/Service… from the Data menu and then choosing WSDL as the service type. This will bring up a dialog box that asks you to provide the URL of the service's WSDL specification:

Like before, Flash Builder 4 will fetch the WSDL file from the server and introspect it, extracting all the available remote procedure calls. You will then be able to drag and drop the data into your application like before.

Accessing XML Services

Of course, SOAP is not the only way of retrieving XML data from a remote location. Flash Builder 4 provides facilities for introspecting a remote URL that simply returns a flat XML document and extracting information from it.
Once again, you will need to click on Connect To Data/Service… from the Data menu and, this time, choose XML as the service type. Flash Builder 4 will ask you to provide the URL of the service you wish to access, invoke it and create a stub class in AS3 to encapsulate the data:



The resulting data provider will become available in the Data/Services panel of Flash Builder 4's GUI, from where you can connect it to your components like before.

Hot Tip Be mindful of the fact that, when manipulation raw XML, Flash Builder 4 has no way of determining whether your service provides data in a consistent format. Therefore, you should ensure that this is the case, or your service call may unexpectedly fail at runtime.

JSON and XML

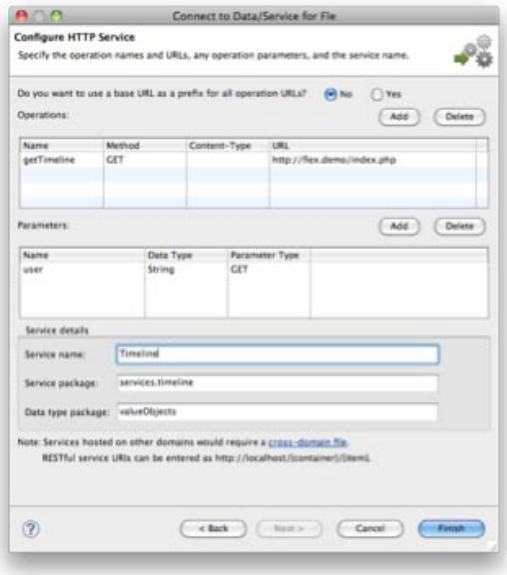
JSON (JavaScript Object Notation) has rapidly become a very popular choice for web service development because of its simplicity, lightweight format and ease of use in a number of languages.

While PHP has had built-in support for JSON since version 5.2.0, AS3 does not have any facilities for manipulating JSON data. Luckily, Flex provides a number of different ways for using JSON.

To start, you will need a PHP script that takes zero or more parameters either through a GET or POST HTTP transaction and outputs JSON-formatted data. For example:

```
<?php

function getTimeline($user) {
  $data = json_decode(
    file_get_contents("
      http://api.twitter.com/1/statuses/user_timeline.
      json?screen_name=" . urlencode($user)));

  foreach($data as $v) {
    unset($v->user->protected);
  }

  return $data;
}

echo json_encode(getTimeline($_GET['user']));
```

The simplest way of connecting to this service consists of once again using the Data/Service Connector wizard to access arbitrary HTTP-based web services. Choosing "HTTP" from the Connect To Data/Service… menu will result in this dialog, where Flash Builder 4 asks for the URL of the service and its parameters:



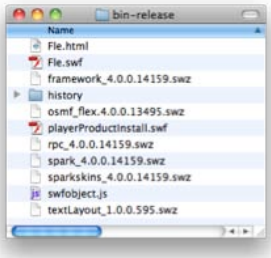
Once you provide the correct information and click on Finish, the wizard will once again create all the infrastructure required to run your service and make it available as before. The HTTP Data/Service Connection wizard also supports XML data.

DEPLOYING YOUR APPLICATION

In most cases, you will want to develop your application in Debug mode. This causes the Flex compiler to add all sorts of useful information that can be used to debugger to help you address any issues that may occur within your application.

However, when it comes time to deploy your application for production, you will want to switch to a Release build so that you can end up with the most compact and efficient codebase possible. You can do so by selecting Export Release Build… from the Build menu.

Exporting a Release build causes a new directory, called bin-release, which contains a number of different files:



Most of these files play a support role to your application—in fact, the only one you will normally interact with is the host HTML file that contains the code required to display your application.

Hot Tip You can change the template used to generate your host HTML file by editing the html-template/index.template.html file in your application's root directory.

Passing Data to Your Flex Application

It is sometimes useful to pass data, like request parameters, to your Flex application as it is being initialized on the client browser.

This can be accomplished by introducing a special parameter in the HTML code that causes the application to be embedded in the web page. In reality, Flex provides a series of convenient wrappers that make the job even easier; if you look inside your HTML template, you will find a portion of code that looks like:

```
var flashvars = {};
…
swfobject.embedSWF(
                "Fle.swf", "flashContent",
                "100%", "100%",
                swfVersionStr, xiSwfUrlStr,
                flashvars, params, attributes);
```

All you need to do is change the content of flashVars to suit your need—that same data will be made available inside the application as the FlexGlobals.topLevelApplication.parameters object, where you can peruse it as needed.

Understanding the Security Model for Flash

When running in the browser, Flash employs a very strict security model that places your code in a sandbox through which all network and disk activities are regulated.

Hot Tip For more information about the Flash security model, read the "Security" section under "Application architecture" in the Flash Builder help online at http//help.adobe.com/en_US/Flex/4.0/UsingSDK/

The sandbox is turned off during debugging—therefore, you don't normally become aware of it until you run your code in production mode and find out that your application cannot access any of its remote data.

Flash supports a number of different sandboxes, depending on what kind of data your application needs to deal with. Most of the time, you will want to use the local-with-networking sandbox, which allows your application to access remote locations, but denies all access to local files.

By default, the sandbox model prevents an application from accessing any resources outside of its own domain, unless that domain specifically grants access with a crossdomain.xml file. Therefore, it is important to remember that you may not be able to access information across different domains.

Hot Tip Adobe AIR applications usually run in the local-trusted sandbox and, therefore, are not subject to connectivity restrictions.

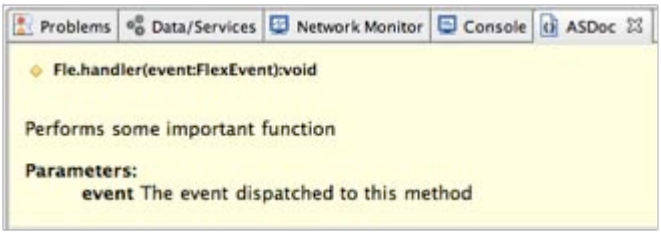
Documenting Your Code with ASDoc

While much of the functionality provided by Flex can be accessed without writing significant amounts of AS3 code, it is entirely possible to address extremely complex tasks entirely within the Flex runtime—systems as complex as encryption engines and image compression libraries have been built in pure AS3 and are used in production every day (in fact, the entire Flex framework itself is built in AS3 as well).

Through a featured called ASDoc, Flash Builder 4 allows you to write inline comments that can be used to document your entire codebase. The syntax used by ASDoc is very similar to the PHPDoc syntax that is commonly used to comment PHP code; for example:

```
/**
 * Performs some important function
 *
 * @param event The event dispatched to this method
 */
protected function handler(event:FlexEvent):void
{
  // Do something
}
```

Flash Builder 4 will automatically scan your code and add any information you write as part of your ASDoc blocks to its code intelligence features; these, in turn, will display the information as you use your code, providing you with a handy dynamic reference for your classes and methods:



ABOUT THE AUTHOR

Marco Tabini is the co-founder, with Arbi Arzoumani, of php|architect, the world's largest PHP magazine in the English language, currently distributed in over 145 countries. He is also the co-founder, with Keith Casey, of Blue Parabola, LLC, a consulting firm that specializes in information architecture, code and security auditing, large-scale deployments and optimization.

An accomplished author on the subject of PHP and the business of web development, Marco is also a frequent speaker at PHP and OSS conferences throughout the world.

RECOMMENDED BOOK

Create desktop applications that behave identically on Windows, Mac OS X, and Linux with Adobe's new Flash Builder 4 platform and this in-depth guide. The book's tutorials and explanations walk you step-by-step through Flash Builder's new, faster tools; the new framework for generating code; how to connect to popular application servers; upgrading from Flex 3; and much more.

BUY NOW
books.dzone.com/books/flex-bible

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: 978-1-934238-96-7
ISBN-10: 1-934238-96-1
50795
9 781934 238967
$7.95