FORMERLY
MX DEVELOPER'S JOURNAL

Web Developer's & Designer's Journal

WWW.WEBDDJ.COM

VOLUME 4 ISSUE 12 **2006**   BUILDING ENGAGING APPLICATIONS AND CONTENT WITH ADOBE TECHNOLOGIES

Ask the Expert

Basics of Flex explained

Plus:

- **Extending a standard Flex ComboBox**
- **Creating Cooler Colors with Adobe's Kuler**
- **Integrating a Flash Interface into Flex 2**

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2007

June 25-26, 2007
New York City

YUM!

INTERSPEC TACULAR

Web Developer's &
Designer's Journal
December 2006

www.frameworksconference.com

FRAMEWORKS

2007 CONFERENCE

Washington D.C.

February 1st & 2nd

New Topics   Great Speakers   Network   Latest technologies

The Frameworks 2007 conference will be located in the Washington DC area, on February 2007. We will have lots of great speakers like last year. This year in addition to Fusebox and FLiP we will have sessions on other frameworks such as Mach-ii, Model Glue, Ruby on Rails, ColdSpring and Struts, and other methodologies such as XP and test-driven development.

We will be looking at frameworks for several web-based languages including ColdFusion, Java and .Net. There will be sessions for beginning and advanced developers, with lots of opportunities for learning from "foreign frameworks" and cross-pollination.

Why should you come to the Frameworks Conference?

"Frameworks is FOR developers, put on BY developers. So the information is useful and there is a ton of it... Are you into "Networking", how would you like to talk to the people who created Fusebox, FLiP, or Fusedoc's? Would you like to meet leaders in our industry using these tools? Well, you can, and that is why you should come."
Eric R. L.

"I'm coming to the conference because our core applications are far behind the Fusebox times–we're running on Fusebox version 2! So I'm interested in learning about other frameworks that are out there, as well as ways of migrating our existing apps smoothly."
Troy B.

"To stay current with the latest fusebox and frameworks conference."
Anne S.

TeraTech
Programming
www.teratech.com

Is XML Overrated?

by Andrew Powell

is XML overrated? This is a question not asked lightly. It is a heavy and bloated question, much like XML itself. XML has been around since 1997. It is document based and it is extremely verbose. It requires a higher payload across the network and cannot be natively used once it arrives. The XML payload must be consumed in some fashion. None of these activities attribute to the speed of an application.

In today's world of AJAX and Flex, is XML past its prime? Does XML need to be regarded as a good idea whose time has come and gone? There are plenty of capable replacements available when using AJAX and Flex. Some examples that come to mind are JSON and AMF. Both of these formats require smaller payloads and less processing on the receiving and sending endpoints. These data interchange formats are faster and less resource intensive than XML.

XML still thrives on the strength of one key factor: its market penetration. As clunky and obtrusive as it may be, XML is still a highly-used standard for data interchange between disparate systems. Most application servers can accept XML and apply some layer of processing to the XML. EDI is a key driver to not only XML's perpetuation, but its very existence.

JSON ("JavaScript Object Notation") is a format that more and more languages are "learning" to consume. It is, as the name implies, a standard object notation. Logic can be created to consume and serialize this notation into language-specific native datatypes. The only limitation to this would be language-specific object instances which cannot be serialized and de-serialized. If more systems were to use JSON for data interchange, in lieu of XML, the payload would decrease and application performance would increase because the parsing of an XML document still outweighs the de-serialization of a JSON string.

Where does this leave the first question? Is XML overrated? There are compelling arguments on both sides of the aisle, but the answer lies in individual preference. If a developer is more comfortable with XML, it will be used. If they are more comfortable with an alternative data interchange format, that format of preference will be used.

Either way, XML will continue to exist; but its days may, indeed, be numbered.

Andy Powell, a member of the editorial board of Web Developer's & Designer's Journal, is a consultant at Universal Mind (www.universalmind.com). andrew.powell@ universalmind.com

Meet Robert

A business executive at a popular social media site

Challenges

- Encountering poor video quality due to exponential growth in traffic.

- Increasing bandwidth costs due to growing audience size.

- Experiencing compatibility issues due to user-generated video arriving in multiple formats.

Solutions

- VitalStream Streaming Services – Improved quality of end-user video experience using a scalable and global content delivery network.

- VitalStream Advertising Services – Transformed the delivery of digital media from a cost center into a profit center.

- VitalStream Transcoding Services – Automatically converted all user-generated content into the leading streaming media format.

Providing End-to-End Solutions for Your Business

VitalStream is more than a rich media delivery company. We are your professional partner providing solutions that meet your unique business challenges. To learn more about VitalStream solutions, call 800-254-7554 or visit www.vitalstream.com/go/solutions/

VitalStream®

Digital Media Solutions for Your Business

SWFAddress –
Deep Linking for Flash

A new open source project
by Rostislav Hristov

SWFAddress is a new open source project that provides deep linking and search engine indexing for Flash Websites and applications. In other words, it enables the Back, Forward, and Reload buttons of the browser and creates unique URLs that can be sent over e-mail or IM. Recently it reached version 1.0 and quickly gained the attention of top design agencies and interactive developers. SWFAddress started as a small experiment that aimed to enhance a typical Flash Website by presenting the portfolio of a fellow photographer. It adopted all the best practices available on the Internet and became one of the best solutions among the growing number of similar libraries. After a successful release candidate, SWFAddress went beyond history management and bookmarking by providing an innovative approach for dealing with search engines. SWFAddress's way to success is based on simplicity, a rich feature set, and quality. SWFAddress is the third free software project of Asual DZZD, a small company from Sofia, Bulgaria. Previously known for its user interface framework, EnFlash, and ActionScript preprocessor, EnLarge, Asual continues to provide new ideas and solutions that target the Adobe Flash Player Platform. All the company's projects rely on the hosting services of Sourceforge. net and take part in the growing community of OSFlash.org. SWFAddress is released under the terms of the open source MIT license.

Technical facts about the project include:
- SWPAddress is a combination of the JavaScript and ActionScript classes that communicate via the ExternalInterface introduced in Flash 8.
- Provides three different ActionScript classes for each version of the language.
- Supports Flash Player 8+ and all the major browsers including Mozilla Firefox, Internet Explorer, Safari, and Opera.
- Uses mod_rewrite with PHP and UrlRewrite with JSP to provide SEO support for Google, MSN, Yahoo and Ask.
- Comes with sample scenarios for Flash, Flex and MTASC.

SWFAddress was the first add-on for the fantastic SWFObject script by Geoff Stearns. Seamless integration with such a popular Flash embedding solution was one of the key values that attracted the developer community. The increased request for SWFAddress will probably lead to support for alternative methods such as the default Flex 2 templates or custom-made Flash writing scripts. SWFAddress can be quickly added to a Website that uses SWFObject only by inserting the JavaScript file like this:

```
<script type="text/javascript"
src="swfobject/swfobject.js"></script>
<script type="text/javascript"
src="swfaddress/swfaddress.js"></
```

```
script>
```

The internal implementation of SWFAddress is simple and straightforward. Once a Flash movie containing the library is embedded in an HTML page, it automatically initializes itself and provides the developer with notification of state changes that need to be implemented. This might not be the easiest task for everybody, but SWFAddress streamlines the process by fully supporting testing workflow inside the Flash Program and Standalone Player. URL changes can be invoked from mouse-click events, frame actions, or HTML links.

Here is sample code that can be added to a MovieClip button:

```
this.onRelease = function () {

    SWFAddress.setValue("/portfolio");
}
```

For links in HTML-formatted TextFields, the following approach should be used:

```
<a href="asfunction:SWFAddress.setVal-
ue,/portfolio/">Go to Portfolio</a>
```

SWFAddress has a single onChange event where the navigation logic needs to be coded. The complexity may vary depending on the Website, but if it's planned from the beginning, it may improve maintainability by centralizing

all the controller actions. A simple navigation handling can look like this:

```
SWFAddress.onChange = function () {

    var navigation = new Object();
    navigation.home = {frame: 1, title:
"Welcome"};
    navigation.about = {frame: 10, title: "About
Us"};
    navigation.contact = {frame: 20, title:
"Contact Us"};

    var value = SWFAddress.getValue();

    website_mc.gotoAndStop(navigation[value].
frame);

    SWFAddress.setTitle(navigation[value].title);
}
```

The setTitle method allows you to add unique titles to every section of the Website and they will appear in the browser's titlebar. It should be used inside the onChange event in order to correctly display the initial title value when the page loads.

The SEO support of SWFAddress is probably the most unique feature of the project. It enables the generation of real links that expose the appropriate content to search engines. The magic happens with some interesting URL rewriting and some trivial server-side code. The following is an example how it works for a Website located at http://www.domain.com that contains a portfolio section.

SWFAddress uses http://www.domain.com/#/portfolio/. The URL that can get indexed is http://www.domain.com/portfolio/ and therefore it should exist as a link in the HTML code of the page. When someone visits the indexed address, the rewriting functionality makes a redirect to the URL that SWFAddress expects. Everything happens transparently for the end user and further navigation through the Website correctly changes the Address bar value. The situation is different when the same link is followed by a search engine bot. In this case the address is rewritten without any redirection to http://www.domain.com/?swfaddress=/portfolio/. The page that handles all this should use a server-side technology in order to generate the appropriate content depending on the "swfaddress" parameter value.

Requests containing GET parameters such as http://www.domain.com/#/portfolio/?desc=true&year=2001 are handled similarly. The search engines will index http://www.domain.com/portfolio/?desc=true&year=2001 and the script will expect http://www.domain.com/?swfaddress=/portfolio/&desc=true&year=2001.

The result is a fully indexed Flash Website with deep linking support.

The following sample queries showcase how the SEO sample hosted at Asual.com is indexed:

```
http://www.google.com/search?q=site:asual.
com+aliquet
http://search.yahoo.com/search?p=site:asual.
com+praesent
http://search.msn.com/results.aspx?q=site:asual.
com+torquent
```

While this approach improves the usability of Flash content over the Web, it also requires an additional effort, such as some manual work for static Websites and some smart code for dynamic ones. The best case will be a Flash CMS that automates the whole process.

Something similar has still not been created for AJAX-heavy Websites, although a part of it is documented in the Unique URLs pattern on AjaxPatterns.org. It will be great for both Flash and AJAX if a technique like this gets adopted and improved over time.

SWFAddress promises to continue adding interesting functionality like AJAX support, transparent user tracking with Google Analytics, and enhanced event handling. It should definitely be considered for your next Flash project. ◢◣

rostislav@asual.com

What Tamarin
Means to Developers

It brings a new level of portability to your existing skill set

by Emmy Huang

On November 7 at the Web 2.0 Conference in San Francisco, we announced that Adobe is contributing source code for the latest ActionScript Virtual Machine (AVM2) to the Mozilla Foundation. AVM2, the new scripting language engine for ActionScript 3.0 introduced with Flash Player 9, was designed to deliver the performance and features to support the needs of rich Internet application developers. The new open source project, known as Tamarin, will be hosted at mozilla.org (http://www.mozilla.org/projects/tamarin/) and contributions will be managed by a governing body of developers from Adobe and Mozilla.

By partnering on the Tamarin open source project, Adobe and Mozilla hope to accelerate the adoption of a standard language for creating rich and engaging Web applications. The Tamarin virtual machine will be implementing the final version of the ECMAScript Edition 4 (ES4) language specification. ECMAScript is a standards-based programming language widely used on the Web – although you are more likely to be familiar with references to Adobe ActionScript, JavaScript, and Microsoft JScript than the name of the foundational standard these languages are based on. Adobe and Mozilla have been working together alongside other companies – such as Yahoo!, Apple, Microsoft, and Opera – on developing the ES4 language proposal as part of the Ecma International Programming Language technical committee (TC39-TG1).

"There is nothing better for a standard than to have it implemented in multiple products," said Secretary General Jan van den Beld of Ecma International. "Adobe is taking a huge step forward in driving standards-based Web development by open source licensing their virtual machine technology."

But What Does This Mean to Me?

As a Web developer, you may be wondering: "What does this mean to me?" For starters, in addition to driving the Adobe ActionScript Virtual Machine in Flash Player, Tamarin will be used in the next generation of SpiderMonkey, the core JavaScript engine in Firefox, and other products based on Mozilla technology. Future versions of Firefox will benefit from a new, high-performance virtual machine for running JavaScript – and applications built with JavaScript will perform faster and more efficiently than ever before. Because it is now available to anyone who wishes to use it, the code behind the lightweight, high-performance ActionScript Virtual Machine and Just-In-Time (JIT) compiler can be used to drive other applications and projects.

"Maybe someone will build a server that relies on the Tamarin virtual machine. Maybe someone will create a client-side product that uses it. Who knows? If you have a project that needs a modern, robust virtual machine, you can use ours," said Sho Kuwamoto, senior director on the Flex Builder engineering team.

The Tamarin project brings a new level of portability to your existing skill set by extending it to other arenas as the use of the virtual machine expands. In the near term, if you have been investing time into learning ActionScript 3.0, you've got a head start to being on the leading edge of the next generation of JavaScript. By unifying the language across browsers and platforms, we also hope to open the Flex and Flash doors to a wider audience of Web developers.

"Developers who know JavaScript will be familiar with the language and syntax of ActionScript," said Mike Chambers, senior product manager for Apollo Developer Relations. "It's going to make it easier for new people to move into Flash and Flex development, and help to expand the Flash ecosystem. Now there isn't a mental barrier that ActionScript is something different, and the learning curve is going to be smaller."

In addition to making your skill set more valuable, standardization around a common language means your work becomes more portable, too. While there are unique extensions, DOMs, and supporting libraries that differ between the browser and Flash Player, you will be able to develop and reuse libraries and algo-

rithms across both the browser and Flash Player. Greater opportunity and more free time – the hardest part will be figuring out which exciting new application you should start building first.

We're Giving to the Community…and Getting Back Even More

We are truly looking forward to contributing to and participating in the open source community through the Tamarin project. This project will improve our ability to maintain a standards-compliant implementation of the ECMAScript 4 specification for ActionScript in future versions of Flash Player, and the virtual machine will benefit from having more smart people looking at and contributing to the code.

"Ideally, this will lead to a more secure, more robust, and a better performing engine for ActionScript and JavaScript," said Adobe engineering director and Tamarin module owner Dan Smith. "Over time, as the virtual machine proliferates into other Mozilla projects, as well as in technologies outside of Mozilla, end users should have a better experience, including better performance and improved security on the client side."

Flash Player isn't the only Adobe product that will benefit from Tamarin. The SpiderMonkey engine is the JavaScript engine within several Adobe products (http://partners.adobe.com/public/developer/opensource/index.html#js), such as Adobe Acrobat, Adobe Reader, Macromedia Director, and Adobe

Shockwave Player. So in a roundabout way, we're sending our technology outside in order to bring it back into our own products!

Ultimately, from Web developers to individuals using the Web, everyone will benefit from having the open source community contribute to the Tamarin code base.

Where to Go from Here

You can learn more about the Tamarin project, read the Tamarin FAQ (http://www.mozilla.org/projects/tamarin/faq.html), and view the source code at the Tamarin project page (http://www.mozilla.org/projects/tamarin) at Mozilla.org. For more information about the Mozilla SpiderMonkey project, visit www.mozilla.org/js/spidermonkey.

To learn more about ActionScript 3.0, see the ActionScript 3.0 Overview (http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html) and visit the ActionScript Technology Center (http://www.adobe.com/devnet/actionscript/) for more language resources. △

Emmy Huang is senior product manager for Flash Player. Her experience includes working in engineering and product management on a range of digital entertainment technologies at Sony Pictures Digital, Liberate Technologies, and Intel. She briefly flirted with the idea of switching to another industry while working towards her MBA at UCLA but decided that working in software was way more fun than selling bleach.

"The Tamarin project brings a new level of portability to your existing skill set by **extending it to other arenas as the use of the virtual machine expands."**

Creating Cooler Colors
with Adobe's Kuler

Adobe believes there is power in color
by Matthew David

Matthew David is the author of multiple Flash books (Flash MX Magic, Building Great Flash MX Games, Flash 3D Bible, and Flash MX Communication Server Bible), has contributed to over a dozen additional books, and writes 300 articles a year for more than 20 international magazines. His area of speciality is next-generation Web solutions. mdavid@matthewdavid.ws

are your colors cool? Do you wish they were cooler? Adobe believes there is power in color. Their new web solution, Kuler, is a clear demonstration of this. In this article, you will find out how to use Kuler to create, share and rate color themes.

Kuler is a new Web-based app from the Adobe Labs. Type in the Web address kuler.adobe.com and you will arrive at the Kuler site.

The focus of Kuler is to allow you to create new color themes. This sounds easy enough. There are other solutions for creating color collections, such as the color palette tools found in some graphics tools through to Web solutions that allow you to convert RGB color to HEX. The problem with all of these tools is that the focus is on creating individual colors. In reality, when you create a print document, Web site or a design what you need is a collection of colors that have a common theme.

Kuler is completely Web-based – a significant first for Adobe, a company that has always created packaged software. It's not clear what inspired the creation of Kuler. The fact that something did spawn this product is good, as we all now have a cool new tool.

Kuler is a designer's dream in that it makes creating custom color groups easy. You can use it to create, share and browse through color collections, so the focus of the site is on color. A traditional color swatch of five colors runs across the top of the screen. The collection of colors is a theme in Kuler jargon. As you might imagine, the whole Kuler solution is in Flash.

The immediate view of the site shows you the highest-rated color themes. Select a new theme and you will see the colors appear across the top along with the name of the color theme, the creator, and tags to help categories the theme.

Along the bottom of each color theme are voting buttons and a reviews people have given any color theme.

Other ways in which you can find colors is to select the Popular and Newest links on the left-hand navigation. The search tool lets you search for colors by tag and name. For instance, entering a search on "holiday" brings back some120 or so themes.

The color theme by itself is very pretty but not much use. What's important is to know what goes into constructing the color. To see the details of any one color you must first select a color, and then choose the "make changes to this theme" button.

The screen changes and takes you into a Details view for any color theme. Along the top of the screen is a color wheel with five highlighted colors. Each color represents a color from the color theme. Each color has the HSV, RGB, CMYK, LAB and HEX values. You can use these values accurately with your designs.

Creating Your Own Themes

Viewing themes others have created is great but how do you create your own themes? If you have an Adobe account then you are halfway there.

There are two ways in which to create a theme: modify an existing theme or create a new theme from scratch.

You are going to learn how to create a new theme from scratch. The principles that you will use for a new theme are very similar to those for modifying a theme.

From the left-hand side navigation, choose the "Create" button. This will give you a basic theme.

The color you want to focus on is the left-hand color. This color is the Base Color and is the pivotal color for your theme. You will see that, by default, the remaining colors fall into a logical color pattern from the Base Color. In addition, notice that the Base Color has a white circle in the color wheel. You can select any of the colors in the color wheel to spread the color pattern.

The default color pattern spread has the label Analogous. You can also choose additional color patterns such as Monochromatic, Triad, Complementary, Shades and Custom.

When you have a color theme you like, you should give it a name. Don't give it a boring name, like "Brown 34," give the color theme a name that's fun such as "Coffee Break." It's also good to give your color theme a few tags to help categorize

the theme. The search engine will use the tags as part of its search.

Finally, select the "Save" button to save your work. (You can select the MyKuler at any time to view all of the color themes you have created.)

Once you have created a theme you can choose to keep the theme private or you can select the "publish" button to share your theme with the whole world.

Kuler Talk

Adobe wants to hear what you think of their latest Web product. Unlike other products coming from Adobe, such as Apollo and Adobe Acrobat 8, Kuler is a pure Web product. The only way you can use it is through your own feedback. To help facilitate open communication Adobe has a Web site you can go to (http://www.adobe.com/cfusion/webforums/forum/categories.cfm?forumid=72&catid=622).

The aptly named "Kuler Talk" is a great place for you to meet other people crazy about color schemes.

To round off your access to information on color treatment and management Adobe has section called Links at the Kuler site. They include:

```
Color Workz - http://www.worqx.com/
color
Color Matters — http://www.colormat-
ters.com/
Wikipedia Color Theory — http://
en.wikipedia.org/wiki/Color_theory
Webexhibits Causes of Color — http://
webexhibits.org/causesofcolor/index.
html
Apple World of Color - http://www.
apple.com/pro/color/
```

Kuler is a fun, focused new solution from Adobe that has potential. What I would like to see in the immediate future are enhancement that allow sample swatches to export to PDF, create RSS feeds for latest colors, and integrate Kuler into products such as Flash, Illustrator and Photoshop.

This is a great first step. I cannot wait to see what will come next.
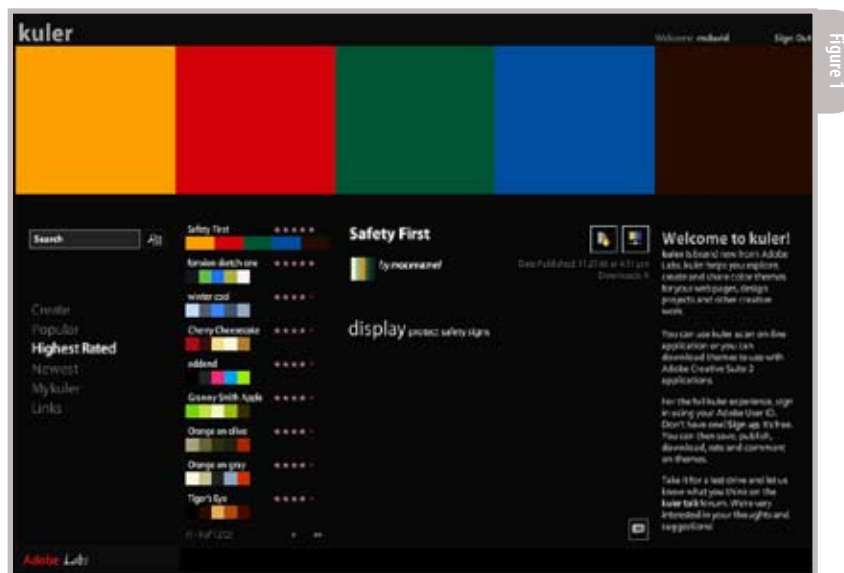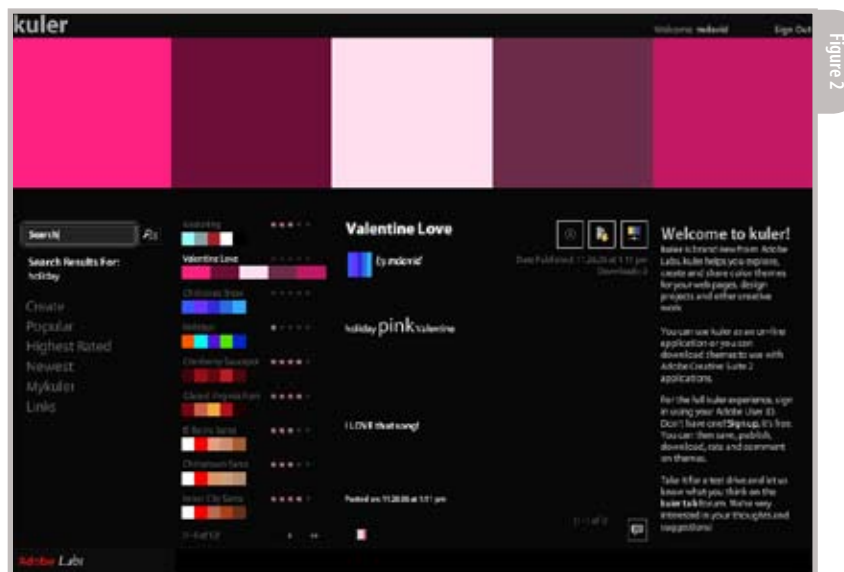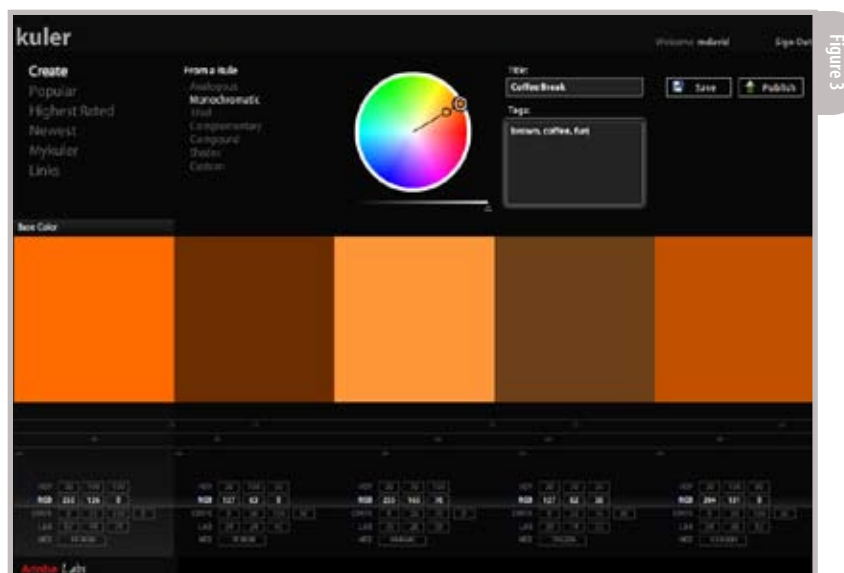
The Value-Aware
ComboBox

Extending a standard Flex ComboBox by adding a missing property to it

by Victor Rasputnis

Dr. Victor Rasputnis is a Managing Principal of Farata Systems. He's responsible for providing architectural design, implementation management and mentoring to companies migrating to XML Internet technologies. He holds a PhD in computer science from the Moscow Institute of Robotics. You can reach him at vrasputnis@ faratasystems.com

What I love about Adobe Flex is that it is a framework in addition to a pretty impressive library of off-the-shelf controls, which can fit the bill for many of the Rich Internet Applications. Flex enables you to create new and extend existing components with a simplicity and elegance hardly ever offered by other GUI development systems. In this article I'll show you how to start extending a standard ComboBox component, which is a combination of edit field, button and a dropdown list. We will be customizing the API and adding some new functionality, making our ComboBox a bit handier than a standard one.

A typical task, while working with a standard ComboBox, is to programmatically select a specific value. Suppose our ComboBox is populated with array of states:

```
    private var usStates:Array=[
        {label:"New York",
data:"NY"},
        {label:"Colorado",
data:"CO"},
        {label:"Texas", data:"TX"
}
    ];
    .   .   .   .   .   .   .   .
    <mx:ComboBox id="cbx_states" dataP
rovider="{usStates}"/>
```

To programmatically select Texas (to the visible portion of the ComboBox), you can write the following index-fetching loop, comparing val against the label of each element of dataProvider:

```
var val:String;

val = 'Texas' ;
for (var i: int = 0; i < cbx.dataPro-
dider.length; i++) {
  if ( val == cbx_states.
dataProvider[i].label) {
     cbx_states.selectedIndex = i;
     break;
  }
}
```

Alternatively, you could look up the data of dataProvider's elements :

```
var val:String;

val = 'TX'  ;
for (var i: int = 0; i < cbx.dataPro-
dider.length; i++) {
  if ( val == cbx.dataProvider[i].
data) {
     cbx_states.selectedIndex = i;
     break;
  }
}
```

Either way these index-fetching loops will clutter the application code instead of simple cbx_states.value='Texas'.

But wait, there is a value property: if a selected object has something in the data property, value refers to data, otherwise value refers to the label:

```
    mx.control.Alert.show(cbx_states.
value); // displays 'NY'
```

Alas, value is a read-only property. It is still helpful as it shields us from selectedItem/ selectedIndex. What we miss is another half and in the following sections we will turn value into a read-write property. That will forever absolve us from index-fetching loops to modify the ComboBox selection.

The simplest way to do this is by extending the original ComboBox so that derived class provides a special setter for the value property. The setter attempts to match the value with either data or label properties of the dataProvider. Once a match is found, it modifies the selectedIndex which should cause the ComboBox to select the matching object:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:ComboBox xmlns:mx="http://www.
adobe.com/2006/mxml"  >
<mx:Script>
<![CDATA[
    public function set value(val:
Object)  : void {
        if ( val != null ) {
            for (var i : int = 0; i
< dataProvider.length; i++) {
                if ( val ==
dataProvider[i].data || val ==
dataProvider[i].label) {
                    selectedIndex
```

```
         = i;
                           return;
              }     }      }
           selectedIndex = -1;
        }
]]>
</mx:Script>
</mx:ComboBox>
```

If the ComboBox.mxml is located under the com/theriabook/controls, its test application can look as in Listing 3 below.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.
adobe.com/2006/mxml" xmlns="*" xmlns:
lib="com.theriabook.controls.*">
    <mx:ArrayCollection id="comboData" >
        <mx:Array>
            <mx:Object label="New York"
data="NY"/>
            <mx:Object
label="Connecticut" data="CT"/>
            <mx:Object label="Illinois"
data="IL"/>
        </mx:Array>
    </mx:ArrayCollection>
    <mx:Label text="Current bound value is
'{cbx_1.value}' " />
    <lib:ComboBox id="cbx_1" value="IL"
width="150" dataProvider="{comboData}"/>
</mx:Application>
```

Run this application, and you'll see the ComboBox displaying the value New York… while we would expect Illinois. We forgot about the order in which objects' properties (cbx_1) get initialized. In particular, the value property is initialized before the dataProvider. And, since during dataProvider initialization ComboBox, by default, selects the first item, the work performed by our value setter is wasted. You can prove the point by just trading places of value and dataProvider in the above application code.

Should we rely on the order of attributes in MXML components? Apparently not. Especially when Flex offers an excellent mechanism to coordinate the updates to multiple properties of the control – the commitProperties() method.

Here is how it works: whenever you need to modify a property raise some indicator, store the value in the temporary variable and call invalidateProperties(), like in the following snippet:

```
    private var candidateValue:Object;
    private var valueDirty:Boolean =
false;
    public function set value(val:Object)
: void {
          candidateValue = val;
          valueDirty = true;
          invalidateProperties();
      }
```

In response to invalidateProperties() Flex will schedule a call of commitProperties() for a later execution, so that all property changes deferred in the above manner can be consolidated in a single place and in the pre-determined order:

```
    override protected function commit-
Properties():void {
        super.commitProperties();

        if (dataProviderDirty)     {
            super.dataProvider = candi-
dateDataProvider;
            dataProviderDirty = false;
        }

        if (valueDirty) {
            applyValue(candidateValue);
            valueDirty = false;
        }
    }
```

Aside from co-ordinating updates to different properties, this coding pattern helps to avoid multiple updates to the same property and, in general, allows setter methods to return faster, improving the overall performance of the control. The entire code of our "value-aware" ComboBox is presented in Listing 4:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:ComboBox xmlns:mx="http://www.adobe.
com/2006/mxml" >
<mx:Script>
    <![CDATA[

    private var candidateValue:Object;
    private var valueDirty:Boolean =
false;
    private var candidateDataProvider:
Object;
    private var dataProviderDirty:Boolean
= false;
```

"Should we rely on the order of attributes **in MXML components? Apparently not.**"

Figure 1

ValueAwareComboBoxDemo - Microsoft Internet Explorer
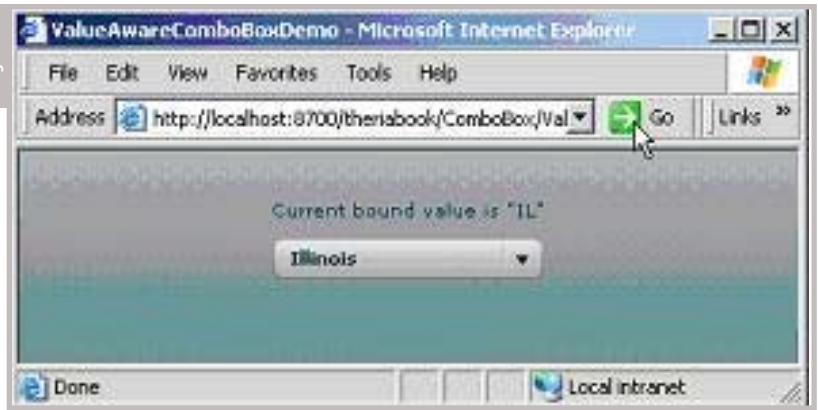
```
    private function applyValue(val:
Object):void {
        if ((val != null) && (data-
Provider != null)) {

        for (var i : int = 0; i
< dataProvider.length; i++) {
            if ( val ==
dataProvider[i].data || val ==
dataProvider[i].label) {
                selectedIndex
= i;

                return;
    }    }    }
        selectedIndex = -1;
    }


    public function set value(val:
Object)  : void {
        candidateValue = val;
        valueDirty = true;
        invalidateProperties();
    }
    override public function set
dataProvider(value:Object):void {
        candidateDataProvider =
value;
        dataProviderDirty = true;
        invalidateProperties();
    }


    override protected function com-
mitProperties():void {
        super.commitProperties();

    if (dataProviderDirty)    {
        super.dataProvider = can-
didateDataProvider;
        dataProviderDirty =
false;
    }

    if (valueDirty) {
```

```
        applyValue(candidateVal
ue);
        valueDirty = false;
    }
  }
  ]]>
</mx:Script>
</mx:ComboBox>
```

Now everything works as expected. The screenshot of the running application is presented Figure 1.

If you change the ComboBox selection, the top label, which initially contains Current bound value is "IL" will change accordingly. No miracles here, a regular Flex data binding one would say. Indeed, good things are easy to take for granted. Still, we have not provided any binding declarations or binding code in our ComboBox. So why does it work? It works because the original Flex definition of value getter ComboBox has already been marked with metadata tag ["Bindable"], which makes the property bindable (you do not have to have a setter to be bindable):

```
[Bindable("change")]
[Bindable("valueCommitted")]
```

But wait, you may say, these binding definitions indicate that data modifications bound to value property get triggered in response to events change or valueCommitted. Yet our value setter does not contain a single dispatchEvent call. Where is the catch? Events are dispatched inside the code that assigns selectedIndex. This assignment results in invocation of selectedIndex setter, which ultimately dispaches events. Remember, Flex is a framework. You have to read the Flex code to take full advantage of it

"This assignment results in invocation of **selectedIndex setter, which ultimately dispaches events.**"
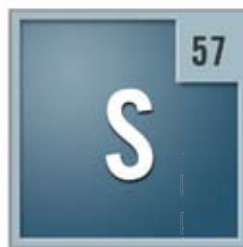
STREAM57

Customized Flash-based media solutions
Software and services for collaboration, video conferencing, and e-learning

I Can't Wait To
Get Started

How I came to give myself more work and love every minute of it

By Paul Mignard

can't wait to get started." The sentiment struck me pretty hard as I struggled to get some sleep on a packed red-eye flight leaving Las Vegas at 11 o'clock at night.

I had visions of components and Flex Forms dancing in my head and I was still reeling from the sheer sense of empowerment that I felt. It's funny, but sitting there with my laptop loaded with the ColdFusion Developers Edition and a copy of Flex Builder, I felt as if I could write the most elegant piece of Web software that my mind could conjure up. I still feel that way now.

But I'm getting ahead of myself.

Simon Horwith, through his blog, recruited me into checking out Adobe MAX 2006 in Las Vegas and report my findings back to you. Having never been, I wasn't exactly sure what to expect. Other than a lot of swag, I anticipated copious demonstrations of Adobe technology and not much else.

I was sorely mistaken.

It all started with CFUnderground on a beautiful Sunday Las Vegas morning. Having no idea what to expect when I stepped into an abandoned bar at 9 o'clock in the morning, I soon relished my time as I made some quick friends and learned something that was true during my entire time at MAX: these developers are some of the nicest and most transparent people I had ever met. Numerous times throughout the day it would not be uncommon for someone to open his laptop and expose his code, either to show you some uncanny way of overcoming some coding roadblock or ask how you could make the code better. People weren't arrogant in the way they explained things and it was welcomed when another developer would join the conversation or even just look over a shoulder to gain understanding. CFUnderground was a fantastic icebreaker, and, for me, was the first time I saw ColdFusion as a scalable object-oriented language. Michael Dinowitz gave a remarkable talk about how to think about ColdFusion in an object-oriented way (which was a helpful precursor to the 23-page article he wrote about the subject that I read just three days later). Mini-MAX soon followed (in the same bar no less) and was a rapid-fire succession of some of the speakers and topics we were going to see at MAX.

Monday morning, after some initial issues with my registration, I was presented with the marvelous MAX 2006 conference bag that would be used by many over the next few days to identify themselves in the throng of Las Vegas gamblers and gastroenterologists who had a much less exciting conference right below ours. I found the exhibit hall to be a great hideout during the pre-conference and in between sessions as developers took root in MAX beanbags and couches strewn all about when they weren't trying to pilfer swag away from the vendors that were strewn about as well. The conference store was stocked (although the Flex books flew off the shelves like they were printed on sheets of $100 bills) and there was always something interesting going on in the small pockets of people gathered around a laptop looking at some chunk of code or wild Flash movie.

Tuesday is when things really began to roll with the first of three keynote sessions. The first keynote was kicked off by an awesome performance from the Blue Man Group and then went on a whirlwind tour of some of the really exciting things that Adobe has up its sleeve. It started with a talk about the marriage of Adobe and Macromedia and gave way to demonstrations of some of the new features in Photoshop, Dreamweaver, Flash, Acrobat, and other things. Ben Forta rocked by showing off the ColdFusion/Flex Wizard in Flex Builder where, in a matter of minutes, he was able to create an entire Flex front-end and ColdFusion back-end to a few tables he had written for a Flex version of iTunes. From there Kevin Lynch took the stage to demo Apollo, the Adobe desktop publishing application. It was cool that it wasn't this new complicated architecture that we'd all have to learn and pore over to get things published. It was a way for us developers to take the stuff we're doing already with ColdFusion and Flex and bring it to the desktop. Needless to say, it started the conference on a good note. Well, that and the $100 million venture

Paul Mignard is an application developer for Liberty University in Lynchburg, VA, where he uses ColdFusion, Flash, and Flex. When not working he is usually trying to keep his blog updated at www.onekidney.com and loves spending time with wife, family, and friends. pjmignard@liberty.edu

fund announced by Adobe to invest in Apollo developers.

The Wednesday keynote focused on Verizon and the ability of U.S. phones to run Flash content now through Brew and the different ways Flash developers can get their content to those phones. The "Can you hear me now?" Verizon guy made an appearance as well as the Flash-enabled Chumby. Thursday was sneak peek day as features were finally revealed from Fireworks to Dreamweaver and, of course, ColdFusion. Ben Forta took the stage aptly dressed as Scorpio Man as he and Tim Buntel showed off the ColdFusion eight-server monitoring tool and the much anticipated cfimage tag. There was some other cool stuff in the form of Soundbooth, the Flex-AJAX bridge, a demo of the ability to export MXML from Fireworks, and a very welcome debugger panel to be found in Flash 9. The keynotes were quite a production as over 3,000 people converged into the giant auditorium each morning in fervent anticipation of hearing the morsels of information Adobe was going to release next.

The developer sessions were something that I would really come to appreciate as being the greatest part about MAX. Gone was the lofty manager-speak and 25-cent developer phrases – here's where we got down to code and, at the risk of sounding like a haughty superficial mind shrink, I really enjoyed watching people who knew what they were talking about teach their chosen craft. Any Google search will return the content of those sessions but certainly not the spirit – I learned so much from the guy sitting next to me or from the bowling-shirt–wearing ColdFusion team member who always seemed eager to answer my constant barrage of what I thought to be annoying and elementary questions. Two really great sessions from this track were the "Component Development to ColdFusion MX 7" session taught by Dave Gallerizzo and "Leveraging ColdFusion Components in ColdFusion" taught by

Ray Camden (who is about as pleasant and approachable as they come). The Flex sessions were simply amazing and as I sat through them I had to bolt myself to the chair so I wouldn't run out to try all of the new effects and techniques that they were covering. The "Leveraging Flex 2 and Flash Player 9 for Truly Cinematic Experiences" session given by Alex Uhlmann was, well, truly cinematic and the "Building Rich Internet Applications with Flex Builder" by Mark Shepherd was as great a teaching session as it was an example of how fast someone who is experienced with Flex and ColdFusion can build a working and impressive application in a brief amount of time. In short, I couldn't take enough notes and I find myself wishing now that I had had more time to attend more sessions – there was simply too much for one person to take in at once.

So here I am back on the plane. It's funny, but as I'm sitting here thinking of the various development projects that I know I have to rework, based on my new understanding of how to employ the various Adobe products I use on a day-to-day basis – I can't help but think of how my users are going to react to the new things

they're about to experience. What are they going to say when I plunk down that first Flex app in front of them and they see how much more quickly and easily they can do their jobs than previously thought. What are they going to say when the time it used to take me to develop a Flash application drops by 75% or when I ask them to download their first Apollo app to use right from their desktop? Although MAX is a slickly produced and well-funded event, it's really the developers who made the experience for me – the networking alone is worth the price of admission. I left knowing that the ColdFusion/Flex/Flash community isn't one of fierce competition and cutthroat politics, but of friends and a general spirit of cooperation. I'll certainly be coming next year – even if Simon doesn't pay my way again.

BEN FORTA is Scorpio Man
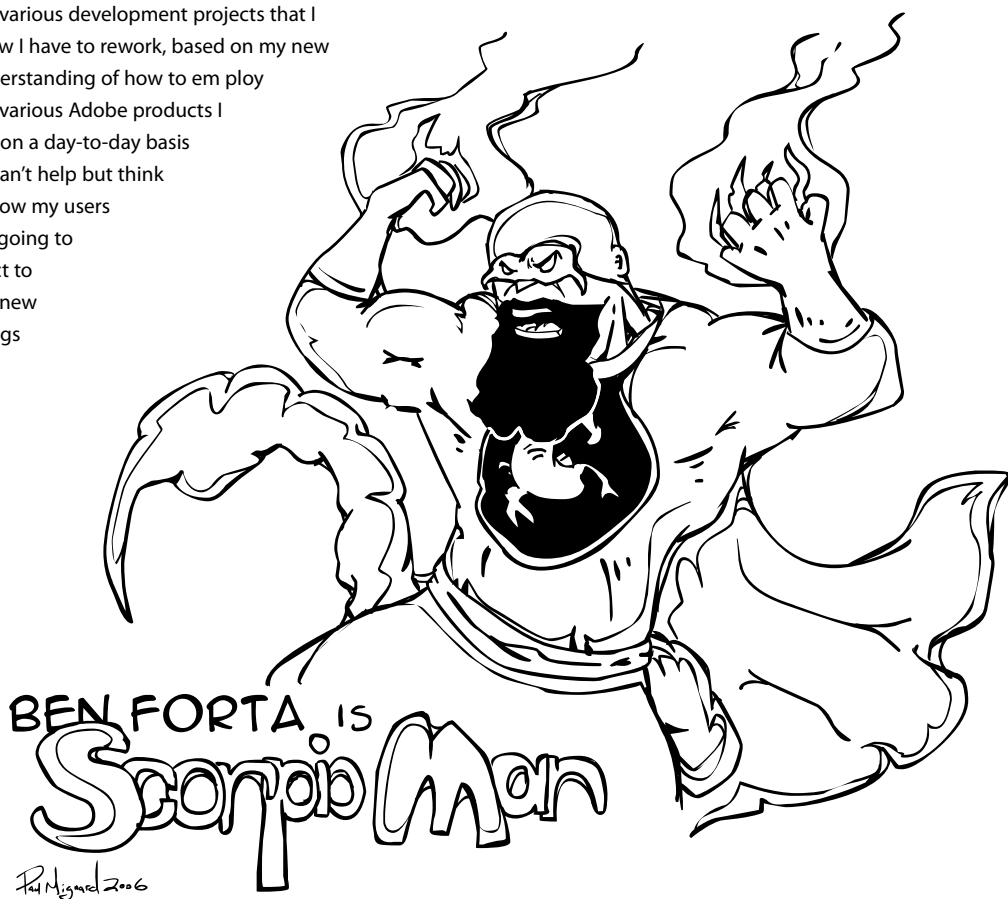
Scary Question.

Exactly who is developing your next app?
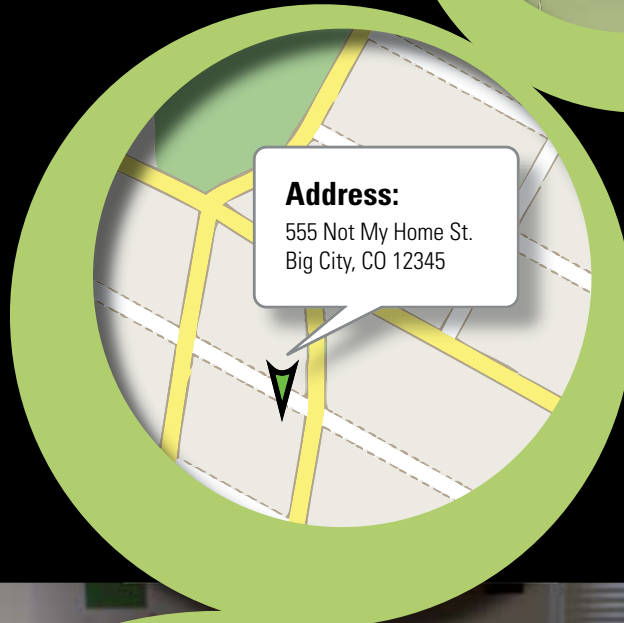
Contact Us

Address:
555 Not My Home St.
Big City, CO 12345

AJAX and
Flash Together

By Mike Potter from
www.riapedia.com

Many people are confused about the relationship between Flash, Flex and AJAX. It's a bit tricky, because in many cases the three technologies are similar to each other: all three allow web developers to add dynamic elements to static web pages very easily. Flex and Flash are both delivered via

the Flash Player (normally used as a plugin to browsers), AJAX applications are delivered via the browser. AJAX is written with JavaScript, XML and HTML components, Flex applications are written with ActionScript, XML and MXML components.

I think that because of these similarities, many people often assume that you'll build your site using AJAX OR Flex OR Flash. The truth is, that's just not the case. There are a number of great examples of sites that are using Flash and AJAX together. Today I'm highlighting two of them: Google Finance (http://finance.google.com/finance?q=ADBE) and Yahoo! Finance (http://finance.yahoo.com/charts#chart1:symbol=adbe;range=1y).

Both sites use Flash to display graphs of stock performance - a popular use of Flash given Flash's rendering engine and performance. What I really like about Google Finance is the fact that the HTML news on the right hand side of the site interacts with the Flash graph on the left. You'll see this when you click a letter in the graph, it will highlight the news on the right hand side (or vice versa).

Google has also written a nice custom Flash component that allows you to change the date range for the graph using a slider, located just below the graph. Even cooler - expanding the graph to show more data adds on more news items (shown as letters in the stock chart), and clicking on a letter not currently shown on the webpage scrolls the HTML news to that item - very nice. Google is using Flash for the graphs, and uses the Flash Java Integration Kit (here it is used on their site: http://finance.google.com/

finance/js/gen_flashelper@sys-con.com, js?v=1.88)

A Flex developer could use the Flex Charting Components in combination with the Flex / Ajax bridge (http://www.adobe.com/devnet/flex/flex_ajax.html) to do something fairly similar but with much less work. (In fact, Nitobi does something similar.)

Yahoo's integration with HTML comes from the left hand "Compare To:" menu item. Clicking on an item in that list will load the stock data into the Flash chart. What I like about the Yahoo implementation is that they'll also change the location in the URL bar for you, allowing you to easily copy and paste a URL so others can see the same view as you are. That allows you to do things like link to this graph showing Adobe's stock compared to Microsoft's over the past year. Yahoo also has a similar time slider to Google - roll over the "Time Range" blue item in the bottom right hand side of the graph to enable it. Notice that the URL bar changes when you select a different time range, allowing me to easily link to the same graph showing Adobe's stock compared to Microsoft's stock since 1986.

The conclusion here should be that the choice of Flash or Flex doesn't need to be an all or nothing choice. You can integrate Flash and Flex into your existing website easily, using technologies like the Flex/AJAX bridge. Adobe has also shown an AJAX bridge to Flex Data Services - I'll get into that early in 2007.

The original version version of this RIApedia post can be found here: www.riapedia.com/2006/12/14/ajax_and_flash_together. Reprinted by kind permission of Mike Potter.

Repositories and Web 2.0

By Andy Powell from
http://efoundations.typepad.com

At a couple of meetings recently the relationship between digital repositories as we currently know them in the education sector and Web 2.0 has been discussed. This happened first at the CETIS Metadata and Digital Repositories SIG meeting in Glasgow that looked at Item Banks, then again at the eBank/R4L/Spectra meeting in London.

In both cases, I found myself asking "What would a Web 2.0 repository look like?" At the Glasgow meeting there was an interesting discussion about the desirability of separating back-end functionality from the front-end user-interface. From a purist point of view, this is very much the approach to take - and its an argument I would have made myself until recently. Let the repository worry about managing the content and let someone (or something) else build the user-interface based on a set of machine-oriented APIs.

Yet what we see in Web 2.0 services is not such a clean separation. What has become the norm is a default user-interface – typically written in AJAX though often using other technologies such as Flash – that is closely integrated into the back-end content of the Web 2.0 service. For example, both Flickr and SlideShare follow this model. Of course, the services also expose an API of some kind (the minimal API being persistent URIs to content and various kinds of RSS feeds) - allowing other services to integrate ("mash") the content and other people to develop their own user-interfaces. But in some cases at least, the public API isn't rich enough to allow me to build my own version of the default user-interface.

More recently, there has been a little thread on the UK list about the mash-ability of digital repositories. However, it struck me that most of that discussion centered on the repository as the locus of mashing - i.e. external stuff is mashed into the repository user-inter-face, based on metadata held in reposi-tory records. There seemed to be little discussion about the mashability of the repository content itself - i.e. where resources held in repositories are able to be easily integrated into external ser-vices.

One of the significant hurdles to mak-ing repository content more mashable is the way that identifiers are assigned to repository content. Firstly, there is currently little coherence in the way that identifiers are assigned to research publications in repositories. This is one of the things we set out to address in the work on the Eprints Application Profile. Secondly, the 'oai' URIs typically assigned to metadata 'items' in the repository are not Web-friendly and do not dereference (i.e. are not resolvable) in any real sense, without every application developer hav-ing to hardcode knowledge about how to dereference them. To make matters worse, the whole notion of what an 'item' is in the OAI-PMH is quite difficult con-ceptually, especially for those new to the protocol.

Digital repositories would be sig-nificantly more usable in the context of Web 2.0 if they used 'http' URIs throughout, and if those URIs were assigned in a more coherent fashion across the range of repositories being developed.

Birth of "Flexzilla"

By Anatole Tartakovsky from
http://flexblog.faratasystems.com

I have been watching browser sta-tistics on our company sites lately - they were quite accurate at predicting the browser usage for 1.5 years ahead of time for the general population - and Mozilla accounts for about of 55% of the hits.

That is the time that "Flexzilla" is sup-posed to be released in the wild. Also, within 9 months, Apollo will be released with similar functionality - joining Web and Flash functionality in a single, small network application platform. Compatibility will be crucial, with an ability to run the same code base, with or without an Apollo extension (discon-nected mode, pocket server, etc.). If done right, performance would further drive adoption rate for Mozilla. And Flash and the XUL platform will give Mozilla neces-sary advantage in terms of richness and performance. That is all for greater good of humans.

What does it mean for developers like you and me?

First, the things that really scare me: JavaScript and ActionScript are not that compatible. The main difference - inter-preted vs compiled really affects the way you code.

It took me 6 months to get used to missing eval() - and in the end in our Flex apps we have an ability to send arbitrary ActionScript to the server, compile it and execute it on the client - within millisec-onds.

It takes a lot of discipline to use strongly typed variables. Put a JavaScript, Flash and Flex developer together and ask them what is typeof x where var x = new Date(); You are going to hear Object, Number and Date - and all of them are right. But if you keep asking "is x + 1 a valid expression?" you will hear Yes from the JavaScript person and No from the ActionScript3 one.

In terms of programming style, you can not live without dynamic closures in JavaScript. I have still to see compatible implementation in the ActionScript. At the end of the day some changes and sacrifices will be made and we will get truly rich dynamic language out of ActionScript. I believe there are some lessons in obscure PowerScript syntax that can be applied here - but I am sure Adobe is looking into it already.

On the bright side, with the release of AS parser, bytecode and minimal code generator it is probably only a matter of time till we would have an evals compiler written completely in ActionScript. We would probably revisit the project I mentioned above and see the cost of full client-side implementa-tion along with bytecode generation (interestingly enough, we started our pre-compiler JavaScript grammar and worked it up to ActionScript one - so we pretty much saw most of the underwater issues there). Anyone interested in participating in opensource implementation of that

please contact me offline.

It's going to be another fun year and a half, with a lot of opportunities and new ideas.

Popup Blocked Warning from Flash
By Danny Patterson
from www.dannypatterson.com

I've noticed recently that some popup blockers will see even the target="_blank" links as a popup. So that got me thinking about a good way to open a popup from Flash and detect if it was actually opened. Rather than participate in the battle against popup blockers I come from the stance that popup blockers are a good thing and we shouldn't try to find ways around them. Therefore, I think the best option is something the just warns the user if a link has been blocked. This is actually very simple with ExternalInterface.

Simply add the following JavaScript function to your page:

```
function openWindow(url, name, width,
height, x, y) {
    var params = "height=" + height +
",width=" + width + ",toolbar=1,scrol
lbars=1,location=1,statusbar=1,menuba
r=1,resizable=1,top=" + x + ",left="
+ y;
    var popup = window.open(url, name,
params);
    return popup != null &&
typeof(popup) != "undefined";
}
```

Then call it from flash like this:

```
var isOpened:Boolean =
ExternalInterface.call("openWindow",
"http://www.google.com", "google",
500, 500, 100, 100);
if(!isOpened) {
    trace("This link was blocked");
}
```

Adobe Flash Player 9 for
Linux Now Available on Adobe.com

By Emmy Huang

From http://weblogs.macromedia.com/emmy Adobe Flash Player 9 for Linux (x86) is now available on the player download center on adobe.com (http://www.adobe.com/go/getflashplayer. This release also includes the standalone and debug players for developers, which are available on the support downloads page (http://www.adobe.com/support/flash-player/downloads.html).

And now for the FAQ

Are the features of Adobe Flash Player 9 for Linux the same as the Windows and Macintosh versions?

Yes, Adobe Flash Player 9 for Linux supports the same major features and functionality as the Windows and Macintosh versions for playback of Flash content and applications. Unsupported features include Express Install and auto-update notification, which are not available due to the variety of Linux platforms, each handling the Adobe Flash Player plug-in installation in different ways. The full-screen viewing feature is not yet available but is expected to be available in a release later this year. For more information about supported features and known issues, please see the release notes http://www.adobe.com/go/flashplayer_releasenotes

Yes, I *did* say at beta that full-screen would make the final release...

So, why isn't the full-screen viewing mode available in Flash Player 9 for Linux?

We wanted to make the new Flash Player for Linux available as soon as possible after the beta release. Unfortunately, the full-screen feature didn't make it into this release but we do expect this feature to be available in the Linux player later this year.

Will WMODE, or windowless mode, be supported with this release of Flash Player 9 for Linux?

No, support for WMODE requires changes to the browser; Adobe is working with Mozilla to enable this functionality for Firefox. For more information and to track the issue, please see https://bugzilla.mozilla.org/show_bug.cgi?id=137189 Bugzilla 137189.

And what about 64-bit support?

No new news to share here. We are working on support for 64-bit platforms as part of our ongoing commitment to the cross-platform compatibility of Adobe Flash Player. We have not yet announced timing or release dates.

Don't forget to check out developing RIAs on Linux with Adobe Flash Player 9, the free Adobe Flex 2 SDK and the free Flex Data Services 2 Express (http://www.adobe.com/products/flex/productinfo/overview/)

Did I mention it was free?! http://the-flashblog.com/?p=240 Adobe

CS3 Icon
3D Carousel

By Lee Brimelow
from http://theflashblog.com/

Let's face it, everything is better when put in a 3D carousel. It's so played out that it's cool.

This little Flash app contains an interactive 3D carousel of all of the new icons for Creative Suite 3. Many people have complained about them but I actually really like the simplicity of them. I mean how much more could they have done with the whole nature/flower concept? I like the whole elements analogy and think it is a nice change. |

There are a couple of icons that are yet to be identified so let me know if you find out. I got most of my information about these from this blog posting (http://stellify.net/visualize-express/adobe-cs3-program-icons-alphabet-soup-on-a-color-wheel-and-what-they-mean) Looking forward to CS3!

Ask the
Expert

Basics of Flex explained
by Andrew Trice

a

ndrew Trice, co-author of
Flex 2 articles on Adobe's
Developer Center, main-
tains a blog at http://www.
cynergysystems.com/blogs/
page/andrewtrice in which
he helps orient newbies and
experts alike about Flex. Web
Developer's & Designer's
Journal is pleased here to
bring a sampler of some
of his more recent shared
insights.

Getters and Setters vs. Public Properties in Flex

I've been asked several times, why would you use get/set functions instead of public variables in your Flex components and classes? Well, there are some great things you can do with getters and setters that you can't do with public variables. On the other hand, there are cases where public variables work better. When using these functions and/or public variables, the code for the caller will be the same:

```
mycomponent.myValue = 1;
```

First, let's look at public variables...

```
[Bindable]
public var myValue : Number
```

It is better to use public variables when there are no additional actions that need to take place when the value has been changed. If you change the value of "myValue", the bindings will update and everything will be handled accordingly. The value will change, and anything bound to that value will change. In this case, there is no need to use getter/setter methods.

Now, on to getters and setters...

```
[Bindable(event="myValueUpdated")]
public function set myValue (value:
Number):void
{
        _myValue = value;
        dispatchEvent( new
FlexEvent( "myValueUpdated" ) );
}

public function get myValue ():Number
{
        return _myValue ;
}

private var _myValue : Number;
```

First I'll explain the [Bindable(event="myValueUpdated")] statement: This indicates that the data binding to the getter's value should be updated when the event of type "myValueUpdated" is dispatched. You'll notice that when the value is set, this event is dispatched, which would notify any components that are bound to this value.

Now, the rest... The code that I showed above doesn't have any benefits over the public property; it functions in exactly the same way, but requires more code. The benefit of getter and setter functions is that they enable sequential code execution when the value is changed. You can create your components so that specific functions are executed any time that the value is accessed using get and/or set functions.

Here's an example:

```
[Bindable(event="myValueUpdated")]
public function set myValue (value:
Number):void
{
        _myValue = value;
        numSets ++;
        myFunction();
        dispatchEvent( new
FlexEvent( "myValueUpdated" ) );
}

public function get myValue ():Number
{
        numGets ++;
        myOtherFunction();
        return _myValue ;
}

private var _myValue : Number;
private var numGets : Number = 0;
private var numSets : Number = 0;
```

In this example, every time the value is set, the numSets Number is incremented, and the myFunction() function is executed. Likewise, every time the value is accessed using the "get" method, the numGets Number is incremented, and the myOtherFunction() function is executed. There is no limit to what kind of code you can execute here. You can have it dispatch custom events, change styles, create new components, etc. This turns out to be very handy when creating custom Flex components.

Object Orientation

Understanding of OOP (object oriented programming) is fundamental in being successful with the Flex framework and being able to get the most out of it. People without a computer science-related (or similar) background may not know much of the fundamental concepts that comprise OOP and how to apply them correctly, so here is a quick piece to help you out.

First, object-oriented programming is a programming paradigm where your code is organized into logical objects, and each object has properties and methods. Each object contains similar and/or related functionality, and is organized into classes that logically represent and logically organize its functionality.

For example: Let's say that we have a class "Automobile." This class would contain the information and functions necessary for our application to use the automobile class. We could have a numeric property for the number of wheels, the speed, and the direction (degrees on a compass). This class would also contain methods that control the actions of the Automobile object: accelerate, decelerate(break), turn, start engine, stop engine, etc. Our class would look something like this...

```
public class Automobile
{
 public var speed : Number;
 public var direction : Number;
 public var numWheels : Number;

 public function Automobile()
 { /* constructor */ }

 public function accellerate() : void
 { /* speed up the automobile */  }

 public function decellerate() : void
 { /* slow down the automobile */  }

 public function turn( direction :
Number ) : void
 { /* turn the automobile */  }

 public function startEngine() : void
 { /* start the automobile engine */
}

 public function stopEngine() : void
 { /* stop the automobile engine */  }
}
```

Andrew Trice is a consultant with Cynergy Systems in Washington, DC, where he specializes in development of Flex-based Rich Internet Applications. Andrew has over 5 years of proven experience in the RIA industry, including application design and development using Flex, Flash, ColdFusion, J2EE and .NET architectures. andrew.trice@cynergy

Ok, now that we have a brief explanation of what object-oriented programming is, we can get into some more aspects of OOP: inheritance and interfaces.

Inheritance is a way to form new objects based on existing objects. When a class inherits from a base class, the new class can utilize public and protected properties and methods from the base class. Inheritance can be used to create different objects that utilize functions within the base class, so that the child classes all utilize the same code base. Inheritance can be used to extend the functionality of existing objects, and inheritance can also be used to override and/or change functionality from the base class.

In ActionScript 3.0, you can access the parent class of your class by using the "super" keyword. For instance, calling the constructor of the parent class would use "super()", where accessing a method of the parent class would use something like: "super.myMethodName()". If a property of the parent class is created with public or protected access, you can access that property in the child class directly by the property name (you would use this.propertyName, not super.propertyName).

Now, let's take our automobile example and apply it to inheritance. We already have a base Automobile class that covers the basic functionality. We can create child classes that extend the functionality of the automobile.

```
public class SportsCar extends
Automobile
{
 public function SportsCar()
 {
   super();
 }
 override public function accelle-
```

```
rate():void
 {
   /* we can override the accellerate
function
        so that it accellerates faster
than the base
        Automobile */
 }
}


public class Truck extends Automobile
{
 public function Truck()
 {
   super();
 }

 public function tow() : void
 {
   /* we can add a tow function that
      allows the Automobile class to
tow
      items. */
 }
}
```

These classes extend the base functionality of the Automobile class, and therefore are instances of the Automobile class. If we have a function outside of the Automobile class, which takes an automobile as the parameter, both a SportsCar and Truck will work since they are both Automobiles. We could have a function such as the following: If we pass in a Truck Instance, and a SportsCar instance, both will work, and each will use the functionality of their specific class instead of the base Automobile class.

```
public function race( auto1 :
Automobile, auto2 : Automobile ) :
void
{
 auto1.accelerate();
 auto2.accelerate();
}
```

I'll get into some more fine-grain details about inheritance later. Now, let's move on to interfaces.

Interfaces are slightly different than inheritance. An interface is a set of "rules" that an object must adhere to. The "rules" are actually method signatures and variable instances that your class (which implements the interface) must implement. When we define an interface, we define method signatures that are required for classes that implement the interface. There is no actual code in an interface; it simply defines methods that must exist within your class. Your class that implements the interface must implement the code for the actual function. If you have multiple classes that implement an interface, those classes must have the same functions (only the ones required to implement the interface), but that is where the similarities of the two classes may stop. They could have completely different logic and properties within them. This is where inheritance and interfaces differ. Two objects that inherit from the same base class have a lot in common (properties and methods); two objects that implement the same interface only have those interface method signatures in common.

Let's now make an Automobile Interface that defines the functions required to create an IAutomobile object (note the "I" stands for "interface"):

```
public interface IAutomobile
{
 function accelerate() : void;
 function decelerate() : void;
 function turn( direction : Number ) :
void;
}
```

We can use the IAutomobile interface to create objects (classes) that behave as

"Inheritance is a way to form new objects based on existing objects"

Automobile objects. These classes do not necessarily inherit from each other and do not necessarily share any common properties.

```
public class Car implements
IAutomobile
{
 private var direction:Number;
 private var speed:Number;

 public function turn(direction:
Number):void
 {
   this.direction = direction;
 }

 public function decellerate():void
 {
   this.speed++;
 }

 public function accellerate():void
 {
   this.speed--;
 }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas implements="IAutomobile"
 xmlns:mx="http://www.adobe.com/2006/
mxml">
 <mx:Script>
  <![CDATA[

   private var direction:Number;
   private var speed:Number;

   public function turn(direction:
Number):void
   {
     this.direction = direction;
   }

   public function decellerate():void
   {
     this.speed++;
   }

   public function accellerate():void
   {
     this.speed--;
   }
  ]]>
 </mx:Script>
</mx:Canvas>=
```

The previous two components both implement the IAutomobile interface, but have nothing else in common. One is simply a class that implements the interface, the other is an MXML component that implements the interface. The MXML component example extends the mx:Canvas component (the same thing could be done by creating an AS class that extends mx.containers.Canvas). Now, let's look at a function similar to the "race" function from earlier:

```
public function race( auto1 :
IAutomobile, auto2 : IAutomobile ) :
void
{
 auto1.accellerate();
 auto2.accellerate();
}
```

This example will work with either object that I have created because both objects implement the IAutomobile interface. The do not rely upon functions in the class hierarchy, just those that were implemented for this interface. You can also use multiple interfaces on classes that you create. Implementing multiple interfaces basically means that you are adding more required method signatures to your class, and you will have to implement these methods to satisfy each interface. On the other hand, you cannot inherit from multiple classes. Some programming languages allow for multiple inheritance (http://en.wikipedia.org/wiki/Multiple_inheritance). ActionScript 3.0 does not support multiple inheritance (so I'll stop there).

Enough of this rambling…what does this have to do with Flex?

Inheritance and interfaces are used extensively in AS3 to create the Flex framework. Just look at the Flex documentation for the mx:Canvas component (http://livedocs.macromedia.com/flex/2/langref/mx/containers/Canvas.html) and you can see inheritance in action (http://livedocs.macromedia.com/flex/2/langref/mx/containers/Canvas.html).

```
Canvas
—>
Container
—>
UIComponent
—>
FlexSprite
—>
Sprite
—>
DisplayObjectContainer
—>
InteractiveObject
—>
DisplayObject
—>
EventDispatcher
—>
```

All Flex framework components that are rendered to the screen extend from the UIComponent class. All combo boxes and lists implement the IList interface, an AbstractService, DataService or EventDispatcher object implements the IEventDispatcher Interface. You may be using these concepts every day, but weren't aware of it, and how you can use it to your own benefit. Inheritance seems easier to take advantage of at first. Let's say that you want to create several objects, all of which will have identical functions and variables. It is easy to see that you can create a base class that encapsulates all of the common functionality. You can then create sub-classes that implement the differing functionality for each class.

When putting these concepts into real-world Flex applications, you'll need to get familiar with the following keywords:

• **extends** (http://livedocs.macromedia.com/flex/2/langref/statements.html#extends): This is used when defining a child class from a parent class.

```
public class A extends public class B
```

• **implements** (http://livedocs.macromedia.com/flex/2/langref/statements.html#implements): This is used when implementing an interface.

```
public class MyClass implements
MyInterface
```

• **final** (http://livedocs.macromedia.com/flex/2/langref/statements. Final

"The benefits of getter and setter functions is that they enable sequential code execution when the value is changed."

methods and properties in OO-languages are generally faster than non-final methods and properties b/c the language does not need to compile declarations that allow the item to be overridden.

```
final function myFunction() : void
```

- **static** (http://livedocs.macromedia.com/flex/2/langref/statements.html#static): This is used when creating variables or functions in a class that are specific to the class, not the instance. Static properties and methods do not require variable instantiation to be executed. Static methods and properties in OO-languages are the fastest thing to access becuase there can only be a single instance of that property or method (and thus there is less for the compiler and runtime to process).

```
public static function
myStaticFunction():
void
//to use it call
it directly from
classMyClass.myS-
taticFunction()
```

- **internal** (http://livedocs.macromedia.com/flex/2/langref/statements.html#internal): This is used when creating a method or property that can be accessed by any object within the same package (namespace).

```
internal var foo : String;
```

- **override** (http://livedocs.macromedia.com/flex/2/langref/statements.html#override): This is used when creating a function that overrides a function within a parent class.

```
override public function myFunction()
: void
```

- **private** (http://livedocs.macromedia.com/flex/2/langref/statements.html#private): This is used when creating methods or properties that are only availaibe to the class where it is defined. A private variable cannot be accessed by outside classes or from descendant classes. Private variables are by nature of OO-languages faster than public vairalbes and shold be used most often.

```
private var myPrivateValue : String;
```

- **protected** (http://livedocs.macromedia.com/flex/2/langref/statements.html#protected): This is used when creating methods or properties that are only availaibe to the class where it is defined and descendant classes. A protected variable cannot be accessed by outside classes.

```
protected var myProtectedValue :
String;
```

- **public** (http://livedocs.macromedia.com/flex/2/langref/statements.html#public): This is used when creating properties and methods that are available to any class.

```
public var myPublicValue : String;
```

www.AjaxWorldExpo.com

AJAX WORLD™ EAST
CONFERENCE & EXPO

NEW YORK CITY

THE ROOSEVELT HOTEL LOCATED AT MADISON & 45th

SYS-CON Events is proud to announce the AjaxWorld East Conference 2007!

The world-beating Conference program will provide developers and IT managers alike with comprehensive information and insight into the biggest paradigm shift in website design, development, and deployment since the invention of the World Wide Web itself a decade ago.

The terms on everyone's lips this year include "AJAX," "Web 2.0" and "Rich Internet Applications." All of these themes play an integral role at AjaxWorld. So, anyone involved with business-critical web applications that recognize the importance of the user experience needs to attend this unique, timely conference – especially the web designers and developers building those experiences, and those who manage them.

BEING HELD MARCH 19 - 21, 2007!

We are interested in receiving original speaking proposals for this event from i-Technology professionals. Speakers will be chosen from the co-existing worlds of both commercial software and open source. Delegates will be interested in learning about a wide range of RIA topics that can help them achieve business value.

Where's i-Technology
Headed in 2007?

at the end of each year, when SYS-CON informally polls its globe-girdling network of software developers, industry executives, commentators, investors, writers, and editors, our question is always the same: where's the industry going next year?

Every time, the answers are surprisingly different from the year before, and of course throw light not just on where the industry is going but also how it's going to get there, why, because of who, within what kind of time-scale – all that good stuff.

Enjoy!

Ruby on Rails • JRuby • AJAX • Rules-Based Programming
JASON BELL
Enterprise Developer, Editorial Board Member, Java Developer's Journal

My predictions for 2006....

1. ***Incremental mainstream adoption of Ruby on Rails.*** It's going to happen, isn't it? Keep an eye out for Sun's offering of JRuby. Whether this is the death of other open source scripting languages like Groovy remains to be seen. Ruby has been a wake-up call and has now drawn the line dividing serious scripting languages from "hobby" languages (ones that wouldn't see enterprise adoption). For me, my job just got a whole lot easier, a whole lot quicker.
2. ***A slowdown in the AJAX hype.*** I think the shine has worn off. There are some

nice applications about but at the end of the day it's a Web page with some very fancy JavaScript.
3. ***2007 is the year of the business rule.*** Rules-based programming will be big business. With the likes of JBoss acquiring Drools it's certainly an area to keep an eye on.

LAMP • REST • ATOM • Apple
DAVID HEINEMEIER HANSSON
Creator of (Ruby on) Rails

1. 2007 will be the year where LAMPers finally decide to stop being neutral about the WS-* mess and pick the side of REST: the next wave of Web APIs will stop supplying both a SOAP and REST API and just go with the latter.
2. On the leading edge, we'll see the same for RSS vs ATOM. For techies in the know, ATOM will become the assumed default syndication format and that'll mark the slow decline of RSS (though more as a technology than as a brand, RSS will remain synonymous with feeds).
3. Apple will continue to trounce everyone else for the preferred geek platform. The stigma of being a Web programmer still using Windows will increase.

Vista • Office 2007 • Zune • AJAX • Ruby • Java • Ruby on Rails • Flash Memory
GARY CORNELL
Founder & Publisher, Apress

In no particular order:
1. IE 7 will have a fast adoption curve and so Firefox will cease gaining market share.
2. Vista will have a slow adoption curve.
3. Office 2007 will have a slower adoption curve.
4. Oh, the Zune will have no adoption curve.
5. The AJAX bandwagon will gain even more speed.
6. Ruby's momentum will slow down as Python and PHP frameworks to combat Rails grow in popularity.
7. The open-sourcing of Java will have no effect whatso-ever on Java's slow decline in favor of dynamic languages (Ruby, Python) and C#.
8. Sales of high powered desktop will slow.
9. Apple will no longer gain market share for its desktops and will stabilize at its current meaningless level.
10. Ultra lightweight notebooks based on flash memory with instant on/off will start coming out in large numbers.

SOA & Web 2.0 • "Outside-In SOA" • Semantic Web • AJAX
DAVID S. LINTHICUM
CEO, The Linthicum Group

1. ***The worlds of SOA and the Web 2.0 blur together.*** While many who think SOA don't think Web 2.0, and many who think Web 2.0 don't think SOA, those days will come to a fast

Jeremy Geelan is Sr. Vice-President, Editorial & Events of SYS-CON Media. He is Conference Chair of the AJAXWorld Conference & Expo series and of the "Real-World Flex" One-Day Seminar series.
jeremy@sys-con.com

end in 2007. So, what does this mean to those standing up SOAs today? It's clear that many of the services we consume and mange going forward will be services that exist outside of the enterprise, such as subscription services from guys like Salesforce.com, or perhaps emerging Web services marketplaces from guys like StrikeIron, Google, Amazon, and others. This is outside-in SOA, in essence reusing a service in an enterprise not created by that enterprise, much as we do today with information on the Web. Thus, those services outside of the enterprise existing on the Internet create a Universal SOA, ready to connect to your enterprise SOA, perhaps providing more value.

2. **The rise of the Semantic Web.** The Semantic Web is the abstract representation of data on the World Wide Web, based on the Resource Description Framework (RDF) standards and other standards. Although this notion has been around for some time, in 2007 it will greatly affect how we design, build, and deploy Web 2.0 applications and SOAs, providing a mechanism to track and leverage application semantics, local and remote.

3. **Enterprise applications continue to move outside the enterprise.** With the success of Salesforce.com and many others, we'll continue to see applications move to the Web includ-

ing accounting, CRM, HR management, logistics, inventory management, etc. While many Global 2000 companies will fight this trend, the success of the younger and more nimble up-starts will drive this movement quickly.

4. **The success of AJAX drives traditional software back to the drawing boards.** With the ability to finally provide dynamic rich content and applications over the Web, traditional software vendors will find that they need new products to play in this new world. Indeed, as Google Mail is giving Microsoft fits, so will other more innovative Web-delivered applications leveraging rich client technology such as AJAX. Entire interfaces will have to be rewritten to support AJAX, and end users will demand that we move away from traditional pump-and-pull HTTP programming.

Mobile AJAX · "Mobile Web 2.0" · SMS · LBS · Flash Lite · On-Device Portals

LUCA PASSANI
Wireless Guru & Technology Evangelist, Openwave

Here are my predictions for 2007:

1. AJAX will still be hyped, but we will still see no mobile AJAX-based killer apps, only proofs of concept.
2. JAVA ME will not gain much more ground. Too fragmented. Games and some other apps. No killer apps

though.
3. What people call "Mobile Web 2.0" is not Web browsing. Saying that mobile and Web will converge is trendy in some environments these days. This is wrong and that's hardly surprising: people buy phones to make calls, not to browse the Web, so why should we expect phones to get so much better at browsing the Web?
4. SMS will still represent 80% or more of data traffic. The rest will be downloads: ringtones, wallpapers and games. WAP will be mostly used as a discovery mechanism to get to those contents. Reformatting proxies to adapt Web content for mobile will be implemented by most operators. They will increase browsing a bit, but nothing earth-shattering.
5. Not sure about Location-Based Services. LBS have been on the verge of explosion for some time now.
6. Flash Lite will make significant progress in Europe and North America, also on operator portals.
7. On-Device Portals are an interesting development: content gets pushed to devices while the user isn't watching and they may decide later to buy it or not. This will be trendy next here. It will be interesting to see which actual implementations of the concept deliver.
8. More people will realize that device fragmentation is one of the main hurdles for mobile.

> "The convergence of BPM and Web 2.0 enables collaborative development and tagging of sub-processes, **establishing a "process folksonomy" where the best processes can evolve organically"**

Flash Memory • AJAX Productivity • Red Hat • Vista • Notebooks • Ubuntu

MARK HINKLE
Editor-in-Chief, Enterprise Open Source Magazine

Here are my predictions:

1. **Flash-bootable PCs** – It's been a long-time coming but laptop PCs will start booting from flash memory. This will make a huge difference in battery life. Intel will lead the way pushing their NAND flash boot memory on a new laptop platform and Apple will be among the first to adopt. The One Laptop per Child initiative will also provide a demonstration of the first zero disk drive PCs albeit small. Devices like this will inspire creativity on higher end models especially as the price of non-volatile memory continues to drop.

2. **New Crop of AJAX Productivity Applications** – While the buzz around AJAX may fade, the number of robust new AJAX-enabled applications will increase. These applications will be built on evolving AJAX frameworks like Dojo and Rico and commercially backed platforms like OpenLazlo. Of course every new start-up will be secretly hoping for Google to make a bid and join the family that has been expanded this year by Writely and Jotspot.

3. **Red Hat Will Become an Acquisition Target** – Someone will make a bid on the #1 Linux vendor. Maybe Oracle who has done a number on the leading Linux vendor with Unbreakable Linux will take advantage of Red Hat's near 52-week low. Uncertainty and ambiguity in the enterprise Linux market will send Red Hat looking for another partner to avoid being swallowed by the DB maker. Maybe IBM will become Red Hat's white knight.

4. **Open Source Everywhere** – More and more companies will open source legacy products and launch new ones under open source licenses. Database vendor Ingres is going to set the standard that other more conservative infrastructure vendors will follow. Look for new open source initiatives from major infrastructure vendors like BMC, VMware, and even Microsoft.

5. **Microsoft Vista Launch Will Boost Sales of Other OSes** –Microsoft's launch of Vista will start to prompt hardware refreshes which can be nothing but good for Apple. Apple already has momentum, Intel hardware, dropping prices and all the tumblers are becoming aligned for it to creep above its measly 5% market share. Linux desktop vendors will likely see a few defectors from the Redmond camp, though big ships turn slowly. Look for Ubuntu to be the Linux desktop distribution of choice.

6. **Half of All New PCs Will Be Notebook**s – PC buyers are buying more notebooks every quarter and sometime in 2007 the number of shipping notebooks will match the number of desktop PCs or come very close.

IT Enabled Services • Web TV • Visual AJAX IDE Microsoft Atlas • Apache XAP

COACH WEI
Founder, Chairman, & CTO, NexaWeb

Here are my submissions:

1. **IT Enabled Services is going to fly high in 2007.** As a result, we will see:
   a. A lot more venture capital investments into IT Enabled Services;
   b. Of course, a lot of startup activities in IT Enabled Services (new company creation, merger and acquisition);
   c. There will be some significant moves made by "traditional, big companies" into IT Enabled Services too. For example, some of the possibilities are:
      i. Massive reality shows on the Web, instead of being on TV. Can you imagine "American Idol" on the Web? Speaking of this, I think highly of Yahoo's initiative into this area, including its recent acquisition of Bix.
      ii. A major entertainment company (NBC, ABC, etc.) fully embracing Web TV.

2. **AJAX grows up** – which means the following are available and useable:
   a. Visual AJAX IDE (solving the ease-of-development issue. Most likely based on Eclipse ATF);
   b. Declarative AJAX Framework (solving the ease-of-development issue. Most likely based on

Microsoft Atlas and Apache XAP);

c.   Adoption of AJAX within less leading-edge enterprises.

d. AJAXWorld Conference overtakes JavaOne conference. JavaOne is being renamed as JavaScriptOne Conference.

3. **Growing adoption of Web 2.0 technologies within the enterprise**

a. Enterprise Mashup Server emerges as a product category.

b. Less leading-edge c ompanies start to adoption Web 2.0 technologies.

4. **The IPO market shows signs of opening up**

a. One or two Web 2.0 companies go public, the majority of the exits are acquisitions.

b. An increase of IPO filings and going public.

WS-BPEL 2.0 • BPM & Web 2.0 • SOA • XSLT • JSON

JOHN EVDEMON
Architect, Microsoft, with the Architecture Strategy Team focusing on BPM and SOA

E.F. Schumacher, a well-known British economist, once wrote: "I cannot predict the wind but I can have my sail ready." With that thought in mind here are ten predictions and hopes to help get your sails ready for 2007:

1. **The WS-BPEL 2.0 specification will finally be approved as an OASIS standard.** Adoption of WS-BPEL will initially be slow, driven by customer demand. BPEL will evolve beyond a "check box requirement" if people begin using it as a foundation for defining process profiles (conceptually similar to how people use WS-Security today). An updated mapping from BPMN to WS-BPEL will also be published.

2. **The convergence of BPM and Web 2.0 begins.** BPM is about improving performance by optimizing key processes. Web 2.0 is about capturing the wisdom of crowds (or as O'Reilly puts it, the architecture of participation). The convergence of BPM and Web 2.0 enables collaborative development and tagging of sub-processes, establishing a "process folksonomy" where the best processes can evolve organically. Collaboration can occur over simple but highly scalable pub/sub mechanisms (like Atom or SSE). Lightweight tools will enable users to model or reuse sub-processes using a broad set of metadata. While this is an exciting opportunity, there are several technical and non-technical issues that must be addressed before this convergence becomes a reality.

3. **Improvements in SOA management and governance**.

Tools, frameworks and platforms will emerge that better enable:

- Defining and enforcing service development guidelines

- Modeling, managing and enforcing operational policies (e.g., security, service level agreements and others)

- Service simulations (what-if scenarios, impact analysis, etc.)

- Modeling and managing service dependencies

- Service provisioning and de-provisioning

- Configuration management

4. **Workflow isn't confined to the datacenter anymore.** Lightweight, extensible frameworks like Windows Workflow Foundation (WF) enable workflow in places where it may not have been previously considered.

5. **Better UI Experiences.** Declarative user interfaces will enable rich user experiences that can be easily modified or extended with simple mechanisms like XSLT. Familiar business applications like Office provide the user interface to back-end line-of-business systems. The line between AJAX-based UIs and rich desktop UIs will blur, enabling users to enjoy both connected and occasionally-connected experiences. Tools and guidance will make building, testing and deploying these composite UI experiences much easier.

6. **A new category of architecture emerges: Software + Services.** It is hard these days to find an architectural concept that is not some how tied to services. The line between Web services, SaaS and traditional applications will blur to the point where the location, contract and hosting of a service are less important than the capabilities exposed by the service.

7. **JSON without AJAX.** We'll start to see more people using JSON to address the XML bloat problem outside of simple AJAX-based applications. The downside is that this may result in more tightly-coupled applications.

8. **Events and states instead of EDI-style messaging.** Lightweight frameworks will empower developers to starting thinking about solutions in terms of event notifications instead of simple messages passing from point A to point B. Hierarchical state machines enable state synchronization across complex, federated processes.

9. **We stop talking about SOA and "just do it."** Sometimes we spend more time arguing about IT trends than actually using them. In 2007 the tools and specifications we need for enterprise-strength, loosely-coupled solutions have finally arrived – it's time to roll up our sleeves and get to work.

10. **IT finally admits that there is no silver bullet.** Every year I hope to see this happen and every year my hopes are crushed by buzzword-of-the-minute hype machines. (Hey I can dream, can't I?)

AJAX Over-use • JSF • Relational Object Mapping • Macs

BILL DUDNEY
Editor-in-Chief, Eclipse Developer's Journal

1. AJAX will continue to gain momentum as folks continue to have the epiphany that Web 1.0 UI is not good for users. Overuse of the technology will be a real problem. JSF will finally start to become a de facto as well as actual standard due to its ease of integration with AJAX.

2. Open Source enablement will continue to be a hot spot for VC investment. I don't think the perfect business model will emerge in '07 though so the market will still remain 'immature.'

3. Java Persistence API will bring relational object mapping to the long tail of the market. Early adopters will be wondering what all the hype is because the technology is so old in their eyes.

4. Macs will continue their 'thought leader' adoption curve. This is not the year they start to penetrate the corporate IT department.

Web Service Orchestration • Web Services Explosion

ADAM KOLAWA
Co-Founder & CEO, Parasoft

1. I anticipate a significant demand for Web service orchestration in the upcoming year, especially in the United States.

   Many organizations now have at least one Web service, and a growing number already have two or more related Web services. Managing multiple related Web services is considerably more challenging than managing the same number of separate, unrelated Web services. To use these related Web services to achieve your business goals, you need to consider how high-level operations pass through the Web services, then determine how to implement this high-level flow— from start to finish. This can be accomplished in two ways:
   – By programmatically coding the application logic required to tie the involved elements together.
   – By using an orchestration tool to direct the flow through the involved elements, which remain separate.

   I predict that the latter method will be the favorite because it is easier.

2. I also expect an explosion of Web services because they are so easy to expose. Once exposed, Web services basically create interfaces which can be reused. This will significantly reduce the amount of code that needs to writ-ten, which will in turn cut the demand for "bare bones" development.

Server Virtualization • Container-Based Hosting • Linux • Rails • Django • Agile Development

BRANDON HARPER
Senior Software Developer at Acxiom Corp.

The top five technology trends I see happening in the New Year are:

1. Server virtualization is just getting started, and will really make itself known in the coming year. Once we start seeing the quad core CPU archi-tectures as a part of standard infra-structure, it really starts making a lot of sense to start deploying and managing servers and applications as virtual entities rather than specific pieces of hardware. This helps manage the cost and pain of software configuration management, take advantage of being able to process many tasks simultane-ously because of hardware support, as well as allows legacy hardware to be retired in favor of applications running on virtual servers.

2. Container-based hosting is the new kid on the block, and will also start making its presence known in the upcoming year. Commonly labeled as "grid" hosting (which is a technical mis-nomer if you understand distributed computing), it essentially claims to be an infinitely scalable hosting platform. This technology still seems to be half-baked at the moment, but you could have said the same thing about Linux ten years ago.

3. People who normally wouldn't use Linux start to explore it and even replace Windows with it permanently. With Vista, Microsoft seems to be mov-ing to a model in which the Windows operating system is a method to police users with DRM and other nonsense rather than provide developers with a good platform on which to use hard-ware, which is what operating systems are really supposed to be. A lot more consumers who haven't noticed this happening in the past will stand up and notice this year.

4. Dynamic languages and frameworks will continue to make leaps in popu-larity and adoption. Given the cur-rent squeeze on technology talent in the US, companies are going to have to learn how to do more with fewer resources. Moving to dynamic languages and frameworks as well as other simplification such as varying Agile software development practices will enable this to take place. I think the obvious leading candidates here are Ruby on Rails and Django.

5. The enterprise will embrace ways to simplify develop-ment by continuing to embrace open source software and Agile Development strategies. While there are a lot of cries to the effect of Ruby on Rails replacing Java, I think that's complete nonsense as Java is a language and Ruby on Rails is a frame-work. Rapid development languages will certainly make some inroads, par-ticularly where heavy tools have been used to build simple applications, Java is still going to be a major part of the service-oriented enterprise for years to come because of the power and tools it provides as well as its industry support.

SaaS • Open Source • Mobile • Enterprise IM Social Networks • China • Virtualization • RFID • Internet Video • Prediction Networks • Intelligent Wikis

The Z Generation CIO
& IT Professional
By Bob Zurek
Director of Advanced Technologies
with IBM Information Integration
Solutions

The Z Generation is typically described as those who are born some-time during the early 2000s and continue to the 2017 time frame. So what will these Generation Z IT Professionals be experienced with. Here's my prediction for the Top Ten characteristics (and this is just the tip of the iceberg):

1. Grew up in the world of SaaS and open source and wonders why you would ever license and install software. If you still needed to install software, then it should be available in the form of open source. Expects all internal projects to be developed using the open source model.

2. Grew up with a mobile technology and wonders how anyone could run a busi-ness without it. Insists everything be available on a highly portable digital device and everyone in the organiza-tion have a device. No exceptions.

3. Grew up knowing how to leverage the power of social networks for the ben-efit of the corporation. This includes the ability to build out these networks and use them to help build new prod-ucts and technologies. Generation Z CIOs will have a huge advantage as they have grown up as participants in many social networks. China will be a big source of these networks. Websites will be built by the Z Generation CIO to invite outsiders in to help build new and innovative products that have yet

to be thought of by the enterprises internal employees.

4. Grew up using Instant Messaging and will insist that the enterprise use IM as a priority over email and that email will only be used if the communications can't be done using the features of the future enterprise IM platform.

5. Will tap into offerings such as TopCoder.com to supplement project teams. There will be a world of competing Topcoder.com like sites where the best coders in the world will be found to solve very complex algorithms and other challenging software projects facing the IT department. China will be a major provider of these teams.

6. Grew up with a complete understanding of the value of virtualization and therefore, their datacenter will be virtualized and the IT operating fabric will be grid-based, tapping the power of external grids of CPU.

7. RFID Everywhere. The Z Generation will be the ones that take RFID to new heights. Everything that is taggable will be tagged and tracked.

8. CIO and Z Generation IT Professionals will leverage the power of Internet video by taking advantage of companies like BrightCove Networks which will bring knowledge workers engaging channels like "The Customer Service Channel", "The Corporate Strategy Channel", "The M&A Channel" and others. I can see companies like Harvard Business Review and others producing content for these channels. Imagine the "The DataCenter Channel." The topics are endless and will be as easy to find as bringing up your favorite search engine. New content will be generated targeted for companies like "The IBM Channel" or the "GE Channel." I would love to see "The Institute of The Future Channel"

9. Intelligent Wikis will be the primary source of knowledge in an enterprise and will eventually do what data warehousing did for business intelligence. Furthermore, new internal employee-generated communities will spin up to voluntarily invent new projects during their off-hours to showcase their creativity that is typically not known by the employer.

10. CIOs will aggressively adopt Prediction Networks as part of the core business strategy to better help the enterprise gauge where everything from sales to new product development will be successful.

Open Source Java • General Public License v2 • GPL v3

Consequences of Open Sourcing Java
by Tony Wasserman
Professor of Software Engineering
Practice at Carnegie Mellon West and
Executive Director of the Center for
Open Source Investigation (COSI)

The open sourcing of Java under the GPL 2 license will have a ripple effect on various technical and business issues in 2007:

• First, people will closely study the Java source code and find one or more serious bugs, at least one of which has been there since the earliest days of Java.

• Second, a real-time systems vendor will fork the source code, as permitted by the GPL, and create a variant that is "tuned" for real-time applications. This step will be the focus of a major debate within the Java community.

• Third, the open-sourcing of Java will have a positive impact on the adoption and use of open source software in general.

• Fourth, the use of the GPL 2 for open-sourcing Java will inhibit the completion and acceptance of the GPL 3 proposal.

Open Sun • iPod Uno • IT2 • Microsoft VAPOR

…And in Other 2007 News
by Richard Monson-Haefel
Award-Winning Author & Senior
Analyst, Burton Group

1. Jonathan Schwartz open-sources Sun Microsystems. In a move that will surprise everyone Sun Microsystems will announce that it will open source its entire company. Sales, marketing, finance, and even operations will be open to the community for anyone to contribute.

2. Apple computer announces the iPod Uno. The size of a match stick with no screen or controls, the iPod Uno plays one song in a constant loop. Despite its limited capabilities, the tiny device becomes an instant hit and a cultural icon.

3. In what is heralded as the seminal article on the subject, Tim Berners-Lee mentions "IT2" Overnight the term morphs into "IT 2.0," spawning thousands of blog entries and press articles, a dozen books, five conferences, and millions of dollars in venture capital. It turns out that the original article, incomprehensible to most readers, was actually another attempt to explain the Semantic Web and the IT2 reference was just a typo.

4. Microsoft will create the first CMO (Chief Marketing Officer) position. The new CMO will immediately change his own title to Chief Command & Control of Packaging Officer (C3PO) and then announce that Vista will be delayed and renamed Microsoft Virtualization Application Program Operating system Reloaded (Microsoft VAPOR). ◢◣

"The enterprise will embrace ways to simplify development by continuing **to embrace open source software and Agile Development strategies**"

Welcome to the Future of Video on the Web!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2007

June 25-26, 2007
The Roosevelt Hotel • New York City

"Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For Sponsorship and Exhibitor Information:
www.iTVcon.com / 201-802-3020 / itvcon@sys-con.com

List of Topics:
> Advertising Models for Video-on-demand (VOD)
> Internet TV Commercials
> Mastering Adobe Flash Video
> How to Harness Open Media Formats (DVB, etc)
> Multicasting
> Extending Internet TV to Windows CE-based Devices
> Live Polling During Webcasts
> Video Press Releases
> Pay-Per-View
> Screencasting
> Video Search & Search Optimization
> Syndication of Video Assets
> V-Blogs & Videoblogging
> Choosing Your PVR
> Product Placement in Video Content
> UK Perspective: BBC's "Dirac Project"
> Case Study: SuperSun, Hong Kong

Welcome to the Future!
Did you already purchase your ".tv" domain name?
You can't afford not to add Internet TV to your Website in 2007!

Just as 2006 was the year of streaming video and the birth of "Internet TV," 2007 is going to be the move beyond infancy toward the long-awaited and complete convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company will now provide video content, not to mention live and interactive video Webinars and on-demand Webcasts, in order to remain competitive.

Internet TV is wide open, it is global, and in true "Web 2.0" spirit it is a direct-to-consumer opportunity. 20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully-featured PVRs, and significant advances in streaming video technologies, Internet TV is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: Internet TV is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and Internet TV will revolutionize television the way we watch it today.

To be part of this change, register for the first Internet TV Conference & Expo 2007 today!

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

SYS-CON EVENTS For more great events visit www.EVENTS.SYS-CON.com

Integrating a Flash
Interface into Flex 2

A proof of concept
by Jesse Warden

Jesse R. Warden, a member of the editorial board of Web Developer's & Designer's Journal, is Flex, Flash, and Flash Lite consultant for Universal Mind. A professional multimedia developer, he maintains a Web site where he writes about technical topics that relate to Flash and Flex. jesterxl@jessewarden.com

my goals here are to integrate http://www.adobe.com/products/flash/ Flash with http://www.adobe.com/products/flex/ Flex, i.e. not just treat it as a design asset tool, but as a contributor to the functionality of the client; using Flash for enhancing the design, and helping reduce transition code which tends to be verbose in Flex.

Below you'll find a sample application that has integration of a complicated Flash design done using many different techniques. I describe the reasons why you would want to work this way, compare with other techniques, and walk through the code and files in my example.

To be clear, this is a proof of concept, and uses a variety of techniques for the purposes of showing developers a variety of ways to do things. It is not meant to indicate all ways to integrate Flash and Flex, nor necessarily the best ways for all occasions. Yes, I did in fact abuse this design for schizzle my nizzle.

Why Integration?

Early on in Flex 2's release, many Flash developers recognized immediately that there were serious interop issues with Flash and Flex. This stems from the Flash Player 9's new AVM (or ActionScript Virtual Machine) that runs the new ActionScript 3.0 programming language. ActionScript 2.0, ActionScript 1.0, and Flash Player 4 scripting on down uses the old AVM engine. For security and various other engineering reasons, the AVMs cannot talk to each other. This results in a SWF written in the Flash 8 IDE not being able to "talk" to a Flex 2 SWF. The need for doing this is that the Flash IDE allows creation of design content that Flex cannot create on its own. Flex 2 is an awesome programming environment compared to Flash, is approachable by developers, and is more geared towards large programming projects vs. multimedia. Therefore, there is much incentive to get them to work together, and not just by using a loadMovie approach (dynamically loading in the Flash created SWF at runtime).

There are various solutions out there. One is do just that, load in the content dynamically by using a SWFLoader component in a Flex 2 project. In the cases of a ActionScript 3.0 only project, you merely use the Flash.display.Loader class. If the content can operate on its own, and/or show design content correctly, this is satisfactory.

Another is to utilize a small set of classes to communicate via LocalConnection. LocalConnection is a class that allows multiple SWFs on the same computer to talk to each other. This helps bridge the AVM gap via viable APIs. This is asynchronous, however, and error checking is faith based.

A third one is to utilize ExternalInterface. Flash Player 8 introduced the ExternalInterface, and API to allow Flash Player to talk, synchronously (a.k.a. to block) with its hosting environment and receive callbacks from those method calls. This not only allows strongly typing those external calls, but hosts can call into the SWF they are hosting as well. In this method an API is used to have the Flash Player 8 SWF that is loaded into a Flex 2 to make a call to ExternalInterface which calls out to JavaScript on the hosting HTML page. The hosting HTML page then forwards that call into the same Flex 2 SWF. The Flex 2 SWF can do the same thing going back since both SWFs can independently register callbacks in JavaScript.

A fourth option is to create the SWF using the Flash 9 Public Alpha up on Adobe Labs (http://labs.adobe.com/technologies/flash9as3preview/). It has the ability to not only use ActionScript 3 in the Flash IDE, but can also produce Flash Player 9 SWFs. One technique in doing this is to have have AS3 in the talk up to Flex who is loading it. There are some small gotchas with "digging" your way up to talk to Flex, but it can be done.

So, why not use one of the above methods? First off, Flash is powerful; it's not just a design asset. So, simply being loaded in isn't good enough for some functionality: Flash and Flex need to talk. Secondly, LocalConnection isn't synchronous, and doesn't have good error checking. This makes debugging long

and frustrating. Third, ExternalInterface uses JavaScript. You've now gone from two languages to three; ActionScript 3.0, ActionScript 2.0 (or 1.0), and JavaScript as well as at least two different tools; Flex Builder for ActionScript 3.0 and JavaScript and Flash 8 for < in compiling when out code specific sound that comment to have you preloading, without via control can so your tag Embed the use if So, example. for mx.core.BitmapAsset and mx.core.SoundAsset not does currently alpha 9 Flash Flex. assets embed are interop with encounter you?ll problems only classes 3 ActionScript same share You future. solution viable a is thus coming? ?is since fourth about said be much there said, That flow. work production of part major software want don?t typically yet, released isn?t 9) (Flash Blaze Fourth,>

What Does Integration Get You?

If I'm making it sound like integration isn't actually the best solution, merely the lesser of the various evils, that's because it is. This is how things are, and even when Flash 9 is released, things won't be much better. Regardless, Flash can produce design content that Flex cannot, and used together the tools are powerful with the right team. Integrating Flash into the developer's work flow rather than as a design afterthought is the best thing to do on larger programming projects – which is typically what Flex is used for. Either that, or the primary skill set on your team is traditional programming, and thus the natural choice is Flex vs. Flash, in which case, Flash is the odd man out, but still a valid contributor.

As mentioned above, you can use the same code. You can write a Flash component in ActionScript 3.0 and use that same component in Flex. You can integrate it into the same Flex project and check it into the same version control. Traditional

programmers will get that good feeling they typically don't get with Flash projects with regards to maintenance.

You can compile the same code in both programs. Since you are using ActionScript 3.0, Flex Builder, mxmlc, and Flash 9 can compile it. Granted, there are caveats with embedding, and it's a bitch. If I knew RegExp, I'm sure I could get an http://ant.apache.org/ Ant script to do it. Either way, you can even create Flash based test cases that you only use to test the Flash symbol class using the new Document class feature. Flex, which typically wraps the symbol class created in Flash in a UIComponent class via composition, can have its own test case.

You can compile once in Flash. Like the http://www.jessewarden.com/archives/2005/04/tried_eclipse_a.html MTASC days of old, mxmlc, the command-line Flex compiler has the ability to do bytecode injection; a.k.a. compiling incrementally. What this means is that you can compile in Flash for the mere sake of "getting the design assets into theSWF." If you later change the code in Flex, it'll merely change the code, but leave the original SWF design assets intact. Since you are embedding the symbol class into a main Flex SWF anyway, it'll compile the assets in the Flash SWF into the main one, and use the updated code you just changed. This is nice because it basically treats the SWFs as little design asset DLLs, so to speak, and uses the most up to date code. Unlike DLLs, the SWFs aren't needed at runtime . It also means that you don't necessarily need Flash to compile the Flex project, which scores major points with some traditional programmers. Third party developers and/or contractors, based on requirements, can deliver just a SWF to some clients or a SWF and a class(es) without having to turn over a FLA. I always give

my source to clients, but I know some people have legal reasons for not being able to do so.

Finally, you can produce more types of components. Since Flash can do some crazy design ideas that Flex can't, or can't quickly, you can integrate a lot of that into components to be used in Flex; 3D exported to FLV integration, various compositing, and other hardcore visualizations.

How Does Integration Work?

Integration starts with coding the component in Flash you wish to use in Flex. Everything is a component; think like this and, no matter how small the asset, you'll be good to go. You write an AS3 class that represents the asset in Flash. Like the AS2 days of old, you make the linkage ID of the symbol in Flash point to your AS3 class. Since everything is a class now in Flash Player 9, your Flash design component asset is too. This class, by my convention, ends in "Symbol". So, if you are creating a glowing button, it's "GlowButtonSymbol". The "Symbol" denotes a Flash-based asset. That way, when you see the class amongst hundreds of others, you can immediately identify it as Flash-specific. Typically, the only thing you would do in Flex Builder to this class is add the Embed tag at the top. Flash IDE will ignore the Embed tag when compiling.

The second step is to create a UIComponent based class in Flex. This class will use your Flash class via composition. You are not required to extend UIComponent; you could extend Container or Canvas for example. A lot of Flash components, however, are design assets using low-level, boilerplate design functionality, so UIComponent usually is sufficient. Everything has to at least extend, or be wrapped in, a

UIComponent. If it's not, Flex will throw an exception.

That's it!

You can take this a little further. I like to create test cases, classes that test to see if the component, and only that component, works. They basically throw it on the stage, and call their exposed methods, if any. This makes debugging a lot faster and a lot less frustrating. It may seem to take longer at the beginning; it does. You'll find, though, that in the long run you spend less time compiling your whole application only to find one stupid thing in your component made everything else blow up.

In Flash, this is just test_GlowButton.as. This class is a Document class that you input into the Document class field in the Flash 9 IDE. It attaches the component and hits some of its methods. I like to use keyboard commands to test methods and such. You can save this in the same Flash folder as well, thus segregating it from the rest of your code base.

In Flex, I basically do the same thing: make an application class, embed the component, and test its methods. Flex Builder 2 makes me keep these on the root folder, though, since they are Application files.

Pitfalls & Frustrations
"Is it that simple?"

No. There are a plethora of gotchas. I'll try to list the most glaring here.

The most annoying is embedding of sound and images in Flex. For example, Flex doesn't have a library. You therefore use embed tags in your Flex code. This associates an external image or sound with a variable, and you can thenceforth treat that variable as a link to the asset. This embeds it in theSWF, so you don't have to worry about preloading, unlike using loadMovie on external images or Sound.loadSound for streaming MP3s. Flex wraps these assets in special classes called "Asset" classes. A little like when you import a sound into the Flash Library and export it with a linkage name. When you embed a Flash

MovieClip symbol in an eternal SWF, it wraps it in a mx.core.MovieClipAsset class. When you embed an image, it wraps it in a mx.core.BitmapAsset class. This is nice. You get strong-typing with embedded assets.

The bad news is, Flash doesn't have access to these classes. I reckon you could use intrinsics, a class that implies "it'll be in the SWF at runtime", and allows Flash to compile. These have to be installed in the Flash 9 classes directory, though; if you put it in Flex's project directory, things could get dazed and confused. If you have to recompile in Flash for whatever reason, you have to comment out the sound code, compile, then undo your comments. Running Flash + Flex Builder at the same time with design intensive files open uses a lot of RAM, thus your comp goes slow, further exacerbating the frustration.

Fonts. Are. The. WORST! Flex 2.0.1, which isn't out yet, apparently has fixed some font importing challenges so you can import more than just polite ttf's. In the past, if you wanted the Flash Player 8 FlashType (a.k.a. Saffron) engine to show really nice-looking fonts, you had to embed them using Flash 8, and then reference that SWF as the font source in the Flex CSS. Either way, it's commonplace to utilize fonts willy nilly in Flash without thinking about what you are embedding. Static text fields, for example, embed fonts by default. There of course is the issue of how they are embedded. Bold or not? Animation anti-alias or readability? Did Flex already embed the font? Who killed JR?

For example, I finally get a build working last night. I then implemented the preloader via the preload attribute on the Application tag. Suddenly my navigation fonts failed to work. Huh? After two hours, I finally got it working again. The Flashpreloader had two static text fields that were using the same font that Flex was using. The difference? The preloader wasn't using the readability version. An issue like this eats away at your productivity and makes you want to start smok-

"Flex 2 is an awesome programming environment compared to Flash, **is approachable by developers, and is more geared towards large programming projects"**

ing again. Fonts have always been hard in Flash; now you really have to micro-manage that stuff.

Flex Builder caching is another gotcha. When you recompile in Flash, you have to refresh your Flex Builder project. I usually refresh the whole thing; I click on the Flex project base folder, and hit F5. This can take as long as the dreaded "Building 0%" you see on the bottom right of Flex Builder / Eclipse. Basically, it's just copying the updated SWF to your bin folder and/or re-compiling it into our main Flex SWF. This can drive you nuts if you are compiling, and fail to see your changes. Refreshing isn't always enough. Sometimes, if you recompiled fonts or bitmap stuff in Flash, an actual Clean project is appropriate via the Project menu. Takes longer, but it works every time. Good sanity check, that is.

Flash doing whack AS3 builds is another problem. It's probably because I have alpha bits, because sometimes, if your class path changes, it'll compile the SWF, but not with the real class. I'm not sure if Flash 9 is caching ABC files (AS3) vs. ASO (AS2) or what, but I'll call methods on the class, and Flash compiles, but at runtime claims the method doesn't exist. Go strong-typing! Anyway, you'll go nuts staring at correct code, when in fact, the problem is your class path. I suffered a lot from this since I transfer code back and forth between my Mac and PC as well as re-factoring twice.

Frame rate really can suffer in Flex. There are a plethora of reasons for this, but I'm having a hard time finding the time to make reproducible test cases to track down the issues. Here are some theories. First, I think Flash is better at producing SWFs that "know" how to preload themselves. So, if you have a long PNG sequence, the SWF will playback well because... well... because Flash is good at doing animations. The same animation in Flex, though, is slow. Now, this isn't as clear cut as that sentence reads. For starters, most of the Flash stuff I use is in states.

States in Flex utilize the Flash Player 9 DisplayList. The DisplayList allows you to have a MovieClip exist, but not be drawn. My guess is, this violates the typical "Hey, there is a long timeline with a lot of assets about to be played, let's preload it." Now, it's "We don't have to draw anything, so just chill...". Thus, it doesn't preload the required assets, and the animation plays slowly.

States have the benefit of removing children that are not applicable to a state you are in. So, in the case of a LoginForm class that has the states of "signIn", "error", and "register", the actual registration fields are not shown. They can actually exist to support databinding, but the actual vector graphics that represents them aren't drawn. This is great.

For Flash animations, this isn't so great. Especially when they are shown, and it now has to play the animation and preload the assets, thus reducing your intended frame rate. That's my theory, anyway.

Worse, states have the nice feature of removing children from the DisplayList that aren't being drawn in the current state. My guess is, Flash Player uses this as an opportunity to run Garbage Collection on the not-being-drawn-assets. Remember, the unofficial ways the Flash Player 8 garbage collector works is a) every 60 seconds and/or b) when RAM usages increase by 20% or more. In the

case of 60 PNGs , each utilizing a meg of RAM (compressed!), that's a ton of memory usage. Naturally, as soon as that animation is removed from the DisplayList, it's a prime candidate for collection. Meaning, the PNGs are no longer in RAM. When the animation is played again, it's slow because the frames have to be loaded into memory yet again. That's my theory too.

For the preload, my fix was pretty simple. You basically make a frame called "asset preload", and throw everything that animation needs on that frame. You can even put it at the front of the animation, in a graphic, alpha'd to 0. This forces Flash Player to throw all the assets in RAM. Since Flash Player is generally good at not running Garbage Collection while animations are going, you can usually be sure your animation will then run well. It did in my tests, anyway. This does not work if your computer itself is running low on RAM. For example, I ran a bunch of these examples at once, and my computers GC + Flash's I guess were running at the same time, because it would freeze during the animations.

I'm not sure what to do – in Flash, this was easy: you just made a MovieClip's visible property false. It's still there, just not being shown, but as soon as you make visible true, it draws immediately. It is being drawn, though, and taking up system resources. The DisplayList does

> ## "Flash can produce design content that Flex cannot, and used together the tools are powerful with the right team"

the same thing with remove child. You can actually have an animation playing, but not in the DisplayList , taking up resources. It's better than visible = false, that's for sure. Either way, my guess is, visible = false is a better safeguard against Garbage Collection unloading your preloaded animation from memory vs. removing it from the DisplayList altogether. This is easy in Flash, hard in Flex if you are using states.

Sound compression is frustrating too. In Flash, there is a multitude of ways to compress it with more codecs. In Flex, it's a little harder. If you are sharing a lot of assets, you may run across the same sound sounding different because Flex compressed it differently. Really annoying. Or, you forgot to re-comment-in your code that you commented out to compile an ebmedded SoundAsset in Flash... thus you never compiled your sound into the SWF. How's your brain's cookie trail?

Resource usage is the biggest kicker. Eclipse 3.1 on my PC came, get this, standard with a max of 40 megs in the Eclipse ini. That's right, 40 megs! Ah, to be a full-time Java developer, and need only that much. When dealing with multimedia content, we Flashdevs need all the RAM we can get, and then some. Especially since Flex Builder can actually render some of those SWFs in real-time in Design View. It actually got so bad at one point, I couldn't even compile: Java 1.5 would collapse on my Windows XP page file faster than Mike Foley, the guy who lives in a trailer down by the river, does on your coffee table. I closed every program to no avail. For the record, I didn't have this problem on my Intel MacBook using the standalone install of Flex Builder. In fact, I gave up looking for the Eclipse.ini on the Flex Builder install (non-plugin) because the performance was acceptable.

That's the worst of the minefield.

The rest really have to do with the new ways Mouse, Keyboard, and focus events mesh with the updated and now built-in EventDispatcher. It's also key to make sure you know exactly how things are working in ActionScript 3. For example, when listening for MouseEvent. CLICK events from your Flash created button, you'll get two. This is normal. By default, it sends one, as does your UIComponent. Since MouseEvent click events bubble, they'll get to whoever's listening. You should intercept it in your UIComponent wrapper, call stopPropagation on the event, create a new event, and dispatch something more meaningful.

Application Example

Flash Design in Flex Application Proof of Concept: http://www.jessewarden. com/flex/flash_flex/source/

Code Walkthrough

Here is a video tutorial walking through the code. In it, you'll see animations created in Flash and then used in Flex in a variety of ways. I've also implemented someCairngorm 2 action with real dynamic data to showcase this isn't just a proof of concept with no substance. Code Walkthrough Video Tutorial: http:// www.jessewarden.com/flex/flash_flex/ captivate/

Source Code

Please note: I cannot give you access to FLAs and fonts because I legally don't own the design. The video tutorial should show you enough about how they are made. The actual code that made them I did include, as well as theSWFs. Therefore, you CAN compile this in Flex using the provided SWFs.

If you want the design, you can purchase it at Templatemonster.com, direct link to the design here: http://www.templatemonster.com/flash-templates/5393. html.

Source Code to Flash Design in Flex App Example - View Source: http://www. jessewarden.com/flex/flash_flex/source/ srcview/" | ZIP: http://www.jessewarden. com/flex/flash_flex/source/srcview/ JXLDownloadsv3.zip.

Other companies in this magazine spent a lot of time on pretty ads. As you can see, we did not. We spent our time hiring the best people and training them to deliver outstanding support for your website. We spent our time building a state of the art datacenter and staffing it with people who care about your website like it's their own. Compassion, respect, credibility, ownership, reliability, "never say no," and exceed expectations are words that describe our service philosophy. From the first time you interact with us, you'll see what a difference it really makes. And you'll also forgive us for not having a pretty ad.

HostMySite.com

WEB HOSTING • MANAGED DEDICATED SERVERS • COLOCATION • VPS • ECOMMERCE • BLOGGING • EMAIL