

Dec 08-10, 2008  
jack.li@insprise.com

## Flex for Java Developers – D1

Company on web: [flex-training.googlecode.com](http://flex-training.googlecode.com)



## About me

- *Marketing stuff*
  - Book author – Prof. Java Native Interfaces with SWT/JFace, Wrox
  - Article writer contributing to IBM DW
  - Consultant to many MNC and gov agencies
  - Dropout of Tsinghua Uni & graduate of NTU
- *Get real*
  - Passionate developer
  - With experience in C, C++, PHP, Java, C# and AS3

Jack Li © 2008 - RIAShanghai.com

## About you

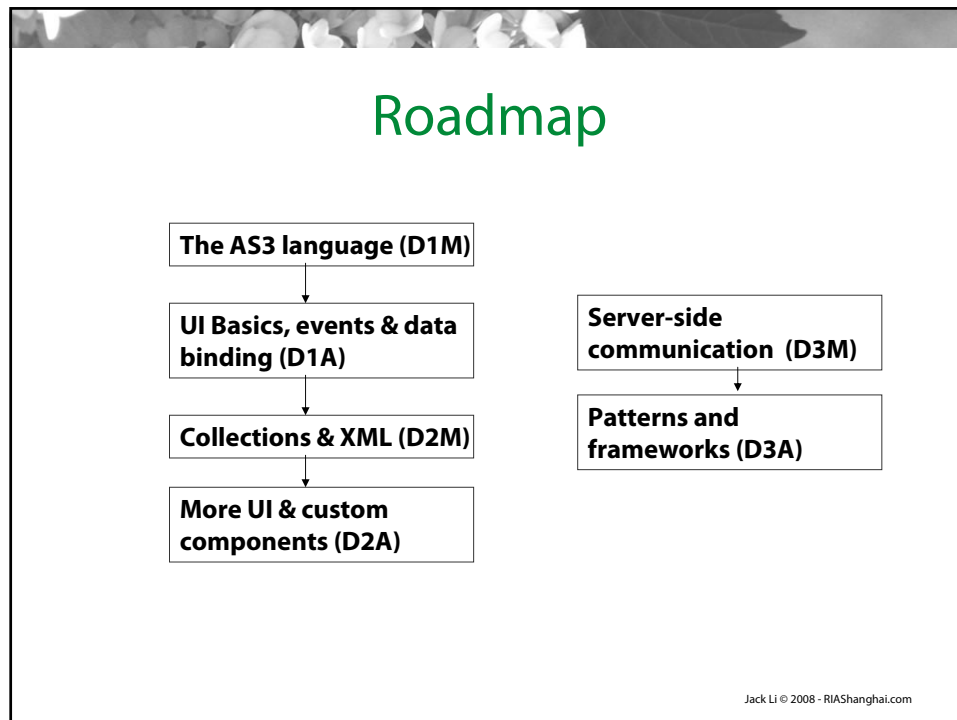
- Who: Developers with Java experience
- Why you are here (a.k.a. objectives): grab the essentials (not superficial) of Flex in 24 hours <mission impossible>
- How: transfer your existing Java knowledge to Flex with the help from the instructor 触类旁通 <mission possible>

Jack Li © 2008 - RIAShanghai.com

## About you (cont'd)

- You are good learners – can easily learn from (e)books and API → topics or answers available on books may not be covered (why?)
- In this training, you'll get:
  - Summarized knowledge instead of plain facts
  - Detailed explanation of important mechanisms instead of superficial usages of certain methods/classes
  - Flex from Java perspective
  - Best practices and patterns from real experience

Jack Li © 2008 - RIAShanghai.com



## For those who self-studied

- Feel free to skip this morning's blah blah bl... if you want.
- Food for thought: the Singleton pattern 单件模式 – 是什么? Java里如何实现? AS3里如何实现? 可参考各类书籍及Internet.
- 编写一个AS3类, 名为Singleton, 来实现这个模式; 并测试.
- Prize: the first one who finishes this task will get a gift or two.



Jack Li © 2008 - RIAShanghai.com

## Flex for beginners



- Flex is a RIA framework implemented in the ActionScript language
- Flash (SWF files) are Flex based applications' binary runtime (Java's JAR's)
- Yes, Flash player is JVM's AS3 counterpart
- ActionScript 3 is a OO language based on ECMA standard

Jack Li © 2008 - RIAShanghai.com

## Sample Flex applications

Jack Li © 2008 - RIAShanghai.com

## Sample ActionScript class

```
package com.test // package is very different from Java's.
{
public class Employee
{
    // Constructor
    public function Employee()
    {
    }

    private var _name:String = "Homer"; // instance variable decl

    public function getName():String {
        return _name;
    }

    public function setName(value:String):void {
        this._name = value;
    }
} // End class
} // End package
```

Jack Li © 2008 - RIAShanghai.com

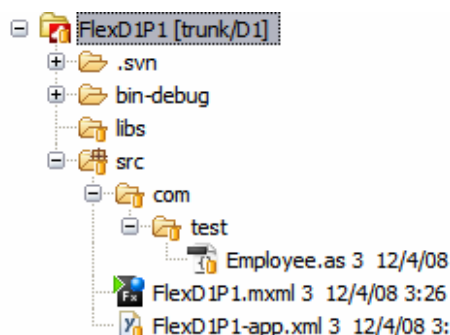
## Your first Flex project

- Open your Flex Builder
- File -> New -> Flex Project, name it: *D1P1*
- Select “Desktop application (runs in Adobe AIR)”
- Once “Finish” is clicked, your first project is created.

Jack Li © 2008 - RIAShanghai.com

## Flex project components

- MXML files – Entry points to applications (Java classes with the *main* method)
- AS Classes
- \*-app.xml – application configuration files



Jack Li © 2008 - RIAShanghai.com

## Entry MXML

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication
  xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute" creationComplete="go();">
<mx:Script>
  <![CDATA[
private function go():void {
  trace("Hello.");
}

  ]]>
</mx:Script>
</mx:WindowedApplication>
```

- *Menu: Run -> Debug D1P1 to run it in debug mode.*

Jack Li © 2008 - RIAShanghai.com

## Your first AS3 Class

- File -> New -> ActionScript class
- Package: **com.test** Name: **Employee**

Jack Li © 2008 - RIAShanghai.com

## Your first AS3 Class (cont'd)

```
package com.test
{
public class Employee
{
    // Constructor
    public function Employee()
    {
    }

    private var _name:String = "Homer"; // instance variable decl

    public function getName():String {
        return _name;
    }

    public function setName(value:String):void {
        this._name = value;
    }

} // End class
} // End package
```

Jack Li © 2008 - RIAShanghai.com

## Revisit the entry MXML

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication
    xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute" creationComplete="go();">
<mx:Script>
    <![CDATA[
private function go():void {
    trace(new Employee().getName());
}

    ]]>
</mx:Script>
</mx:WindowedApplication>
```

- *Menu: Run -> Debug D1P1 to run it in debug mode.*

Jack Li © 2008 - RIAShanghai.com

## When you need help

- The API
- Adobe's documentation in PDF
- Flex Builder's help

Jack Li © 2008 - RIAShanghai.com

## Elements of programming languages

- Variables
  - Methods/Functions
  - Classes
  - Interfaces
- 
- *AS3 v.s. Java v.s. C# v.s. PHP v.s ...: **syntaxes vary across languages but the concepts are the same.***

Jack Li © 2008 - RIAShanghai.com

## Variable declaration

- Different types of variables: class variables; instance variables; local variables in methods.
- Syntax: **[public] [static] var name:Type [= ...]**
- Variable naming follows the same rule as in Java (identifier)
- ```
public class Test {  
    public static var a:int = 10;  
    private var b:String = "Hello";  
    public function run():void {  
        var b:Number = 5.0;  
        ...  
    }  
}
```

Jack Li © 2008 - RIAShanghai.com

## Primitive types

| AS 3        | Java                                           |
|-------------|------------------------------------------------|
| int, <uint> | int <long, short, byte, char>                  |
| Number      | double <float>                                 |
| Boolean     | boolean                                        |
| String      | <i>java.lang.String (not a primitive type)</i> |

<> indicates no corresponding type in the other language

Jack Li © 2008 - RIAShanghai.com

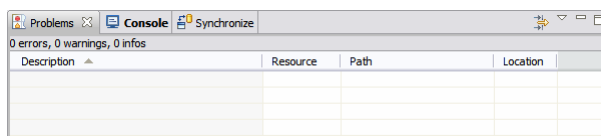
## Numbers

- `var a1:int = 8; a2:int = 0xFF;`
- `var b:int = parseInt("100"); // string to int`
- `var c1:Number = 1.0; c2 = Infinity; c3 = NaN;`
- `if(isNaN(c)) {...} else if (isFinite()) { ...}`
- `var d:Number = Number((100 + 1) + "0"); //  
String to Number - 1010.00`

Jack Li © 2008 - RIAShanghai.com

## Your turn

- Add an instance variable `实例变量` named `_birthYear` into the Employee class; its type is `int`; and it is `private`.
- Hit "Ctrl+S" to save and make sure there is no error:



Jack Li © 2008 - RIAShanghai.com

## Strings

- `var s:String = "hello"; s = new String("hello");`
- `t = "h" + 2; // concatenation: "h2"`
- `s.substr(1, 2); // "el"`
- `s.indexOf("el"); // 1`
- `var len:int = s.length; // length is a property`
- `var c:String = s.charAt(1); // "e"`
- `s.toLowerCase(); s.toUpperCase();`

Jack Li © 2008 - RIAShanghai.com

## Your turn

- Add an instance variable 实例变量 named `_address` into the Employee class; its type is `String`; and it is *private*.
- Hit "Ctrl+S" to save and make sure there is no error.

Jack Li © 2008 - RIAShanghai.com

## Access modifier

- Applicable to all programming elements

| Modifier           | Visibility – ActionScript         | Visibility – Java                                |
|--------------------|-----------------------------------|--------------------------------------------------|
| private            | Same class                        | Same class                                       |
| <b>protected</b>   | <b>Same class and sub-classes</b> | <b>Same class, sub-classes and same package</b>  |
| Internal (default) | Same class and same package       | <i>No modifier</i> - Same class and same package |
| public             | Everywhere                        | Everywhere                                       |

Jack Li © 2008 - RIAShanghai.com

## Functions

- AS3 is loose as it tries to maintain compatibility with old versions.
- Functions can be anywhere – in packages, classes, even in functions.
- As a good practice, use functions as you do in Java – direct members of interfaces, classes only.

Jack Li © 2008 - RIAShanghai.com

## Function signature

- **[public] function funcName(param:Type, [moreParams:Type]\*):Type**

```
// in the Employee class:  
public function setName(value:String):void {  
    this._name = value;  
}
```

Jack Li © 2008 - RIAShanghai.com

## Your turn

- Add a function named *getBirthYear* without any parameter; its return type is *int*.
- In the body of the function, please insert the following:

```
{  
    return _birthYear;  
}
```

Jack Li © 2008 - RIAShanghai.com

## Function closure

```
public class Employee
{
    ...

    public function getName():String {
        return _name;
    }

    // Returns the function that retrieves the name
    public function getNameFunction():Function {
        return getName;
    }
} // End class
} // End package

var func:Function = new Employee().getNameFunction();
trace(func());
```

**A function bound to  
an object instance.  
"function instance"**

Jack Li © 2008 - RIAShanghai.com

## Constructors

- Constructors in AS3 behave the same as in Java except the following:
  - Only one constructor is allowed per class
  - Constructors can not be private

```
public class Employee
{
    // Constructor
    public function Employee(name:String = null) {
        setName(name);
    }
}

var emp:Employee = new Employee("Homer");
var emp:Employee = new Employee(); // default null
```

Jack Li © 2008 - RIAShanghai.com

## Property access styles (getters and setters)

- *Property access style 1 – straightforward*
- Read: `var s:String = employee.getName();`
- Write: `employee.setName(s);`
  
- *Property access style 2 - natural*
- Read: **`var s:String = employee.name;`**
- Write: **`employee.name = s;`**

Jack Li © 2008 - RIAShanghai.com

## Getters & setters, the natural way

```
// public function getName():String {  
//     return _name;  
// }  
//  
// public function setName(value:String):void {  
//     this._name = value;  
// }  
  
public function get name():String {  
    return _name;  
}  
  
public function set name(value:String):void {  
    this._name = value;  
}
```

Jack Li © 2008 - RIAShanghai.com

## Properties

- A **property** must have at least one of getter and setter; (read-only/write-only)
- Getters and setters are still functions;
- You can override getters and setters when subclassing
- Additionally, **any public instance variable is a property:**  
public var email:String;

Jack Li © 2008 - RIAShanghai.com

## Your turn

- Add a getter and setter function for the birthYear property.

Jack Li © 2008 - RIAShanghai.com

## OO in AS3

- Abstraction - interfaces
- Inheritance, polymorphism – subclasses

Jack Li © 2008 - RIAShanghai.com

## Interfaces

```
package com.test
{
  public interface IPerson
  {
    // 1. Declarations of variables are not allowed.

    function get name():String; // 2. no public
    function set name(value:String):void;

  } // End class
} // End package
```

Jack Li © 2008 - RIAShanghai.com

## Implementing an interface

```
package com.test
{
public class Employee implements IPerson
{
    ...
    public function get name():String {
        return _name;
    }

    public function set name(value:String):void {
        this._name = value;
    }
    ...
}
```

Jack Li © 2008 - RIAShanghai.com

## Extending a class

```
package com.test
{
public class Manager extends Employee
{
    public function Manager(name:String=null) {
        super(name);
    }

    override public function get name():String {
        return "Manager: " + super.name;
    }

} // End class
} // End package
```

Jack Li © 2008 - RIAShanghai.com

## Type checking and casting

- `var e:Employee = new Employee();`
- `if(e is IPerson) ...`
- `if(null is Type) // always false`
- `var m:Manager = e as Manager; // m will be null unless e is not null and e is an instance of Manager`
- `var any:*; var any:Object; // any can be anything`

Jack Li © 2008 - RIAShanghai.com

## The Object class

- In AS3, the Object class is at the root of the class hierarchy
- Two important methods:
  - `Object.hasOwnProperty(name:String):Boolean`
  - `toString():String`

Jack Li © 2008 - RIAShanghai.com

## Equality

- For primitive types (String, int, Number, boolean): value comparison '==' '==='
- For object types: reference comparison

Jack Li © 2008 - RIAShanghai.com

## Error (a.k.a. Exception)

```
public class Employee implements IPerson
{
    ...
    public function set
        birthYear(value: int): void {
        if(value < 1900) {
            throw new Error(
                "Invalid birth year: " + value);
        }
        this._birthYear = value;
    }
}
```

Jack Li © 2008 - RIAShanghai.com

## Error handling

- In Java, you have checked exception (IOException) and unchecked (NullPointerException). In AS3, all Errors are unchecked.

```
try {  
    new Employee().birthYear = 1800;  
}catch(e: Error) {  
    trace(e);  
}  
// Console: Error: Invalid birth year: 1800
```

Jack Li © 2008 - RIAShanghai.com

## Your turn

- Please add a getter for property named *age* into the Employee class
- Hint:
  - You need current year to calculate age;
  - What possible class name that may give you the current year? (No idea? Think how you do that in Java) Find it in the API document.

Jack Li © 2008 - RIAShanghai.com

## UI works

- Coding demonstration

Jack Li © 2008 - RIAShanghai.com

## Your code is executed in single-threaded mode.

- The biggest difference with Java
- You can not create, manipulate, or destroy any threads
- The secrete why Flash animations runs much smoother than Java applets
- Max allowed execution time for code in a listener is 15s
- “Your code” but not Adobe’s, e.g. downloads

Jack Li © 2008 - RIAShanghai.com

## Event mechanism

- Participants
  - Event dispatcher or event source
  - The Event object
  - Listeners
- Listeners are registered to event source; when an event occurs at the source, the source will dispatch the event to all registered listeners.

Jack Li © 2008 - RIAShanghai.com

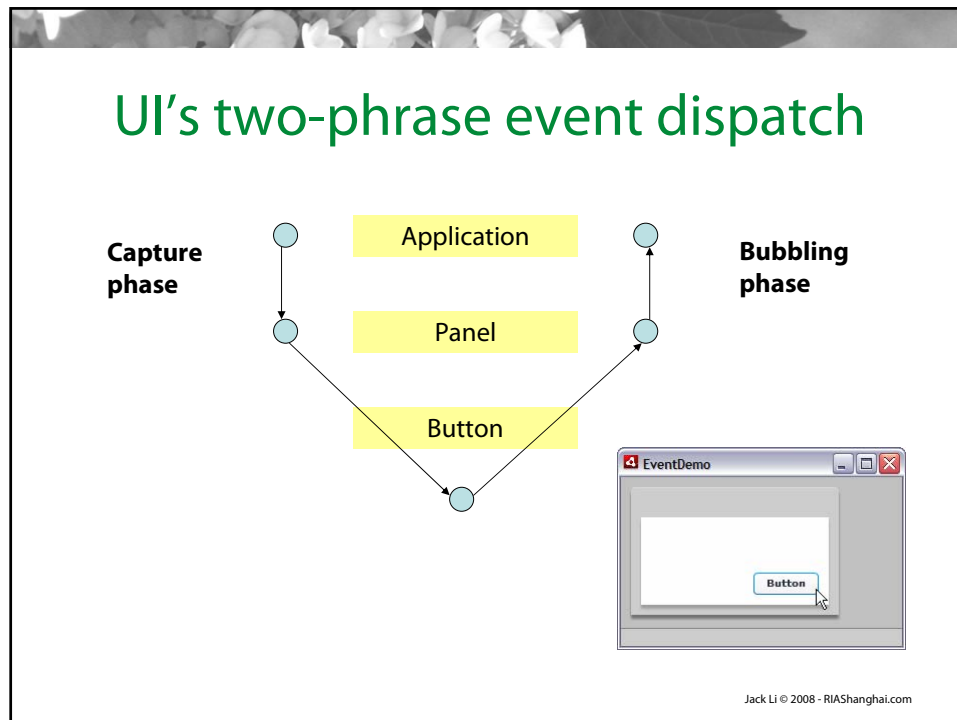
## Example

```
<mx:Button label="Button" id="button"/>
// event handling function must have the following
signature:
private function onEvent(event:Event):void {
    trace(event.currentTarget + " - " + event);
}

button.addEventListener(MouseEvent.CLICK, onEvent);

// button - [MouseEvent type="click" bubbles=true
cancelable=false eventPhase=2 localX=38 localY=10
stageX=229 stageY=165 relatedObject=null
ctrlKey=false altKey=false shiftKey=false
buttonDown=false delta=0 commandKey=false
controlKey=false clickCount=0]
```

Jack Li © 2008 - RIAShanghai.com



## flash.events.Event

- The super class of all Event classes
- Properties:
  - target:Object
  - currentTarget:Object
  - type:String

Jack Li © 2008 - RIAShanghai.com

## Custom event class

- EmployeeInfoChange extends Event

Jack Li © 2008 - RIAShanghai.com

## Dispatching event

- Employee implements IEventDispatcher
- Or simply Employee extends EventDispatcher

Jack Li © 2008 - RIAShanghai.com

## Binding

- In many cases, we need to tie the UI controls with the a property of an object – i.e., the UI should display the value of the property and when the user enters input through the UI the model object should be updated too.

Jack Li © 2008 - RIAShanghai.com

## Making properties bindable

- Making all properties in a class bindable by putting the [Bindable] tag above the class declaration
- Making individual properties bindable by putting the [Bindable] tag above the property declarations

Jack Li © 2008 - RIAShanghai.com

## [Bindable]

```
public class Person extends EventDispatcher {
    [Bindable] // automatically generate PropertyChangeEvent
    public var name: String;

    public var address: String; // not bindable.

    [Bindable(event="ageChange")] // must manual dispatch.
    public function get age(): int {
        return _age;
    }

    public function set age(value: int): void {
        if(this._age == value) {
            return;
        }
        this._age = value;
        dispatchEvent(new Event("ageChange"));
    }
}
```

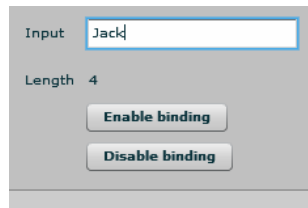
Jack Li © 2008 - RIAShanghai.com

## Setup bindings

- Using {} in MXML
- Or in AS3

Jack Li © 2008 - RIAShanghai.com

## Setup bindings using AS



The screenshot shows a user interface with three main components: a text input field containing the text 'Jack', a label below it that reads 'Length 4', and two buttons positioned below the label. The top button is labeled 'Enable binding' and the bottom button is labeled 'Disable binding'. The entire interface is set against a light gray background.

```
<mx:TextInput id="textInput" />
<mx:Label id="labelLength" />

<mx:Button label="Enable binding"
id="buttonEnableBinding"
click="enableBinding();" />

<mx:Button label="Disable binding"
id="buttonDisableBinding"
click="disableBinding();" />
```

```
var watcherLength: ChangeWatcher;

private function enableBinding(): void {
    if(watcherLength == null) {
        watcherLength = BindingUtils.bindProperty(labelLength,
"text", textInput, ["text", "length"]);
    }
    trace("Binding enabled. ");
}
```

Jack Li © 2008 - RIAShanghai.com

## Stop bindings

```
private function disableBinding(): void {
    if(watcherBindLength != null) {
        watcherBindLength.unwatch(); // calls
        watcherBindLength.reset(null);
        watcherBindLength = null;
    }
    trace("Binding disabled. ");
}
```

Jack Li © 2008 - RIAShanghai.com

## Food for thought

- Why bi-directional binding won't cause deadlock?
- Adobe claims "Flex ensures that bidirectional data bindings do not result in an infinite loop; that is, Flex ensures that a bidirectional data binding is triggered only once when either source property is modified."
- Is that true? How do they do that?

Jack Li © 2008 - RIAShanghai.com

## Extra topics

Jack Li © 2008 - RIAShanghai.com

## Coercion

- Is it magic that you can bind an integer property to the *text* property (String) of a *TextInput* bi-directionally? What happen if the user inputs value like "13a"?
- Compilation-time error, runtime error or no error at all?  
employee["name"] = false;  
employee["birthYear"] = "It's a secret";

Jack Li © 2008 - RIAShanghai.com

## Type annotation

- In Java, you must specify the type when you declare a variable. In AS3, the type specification is optional. If you do, it's a type annotation, e.g. var name:String
- You use '\*' to indicate a variable is untyped – i.e., may contain value of any types.
- The compiler can use the type information to perform checking.

Jack Li © 2008 - RIAShanghai.com

## Compiler's standard mode

- When you compile code in standard mode (not strict), type checking is ignored.

```
var i:int = "13a";  
var dog:Dog = new Apple();
```

Jack Li © 2008 - RIAShanghai.com

## Coercion rule

- At runtime, if the type of the value assigned to a variable and the type of the variable do not match in OO:
  - If the type of the var is primitive, AVM use int(), Boolean(), Number(), String() for conversion. No exception will be thrown.
  - TypeError will be thrown for non-primitive types.

Jack Li © 2008 - RIAShanghai.com

## Coercion instructions

- Related AVM2 Instructions:  
0x80 *coerce(index)* – *coerce a value to any specified type*  
0x82 *coerce\_a()* – *indicates the value on the stack is untyped*

Jack Li © 2008 - RIAShanghai.com

# Premium Flex Training now in CHINA

*The fast way to acquire solid Flex skills - Flex優質培訓*

TRUSTED BY FORTUNE 500  
COMPANIES AND GOVERN-  
MENT AGENCIES. WHAT  
TRAINEES SAID: "THE  
BEST FLEX TRAINING FOR  
JAVA DEVELOPERS" "REAL  
WORLD ACTIONSCRIPT  
TRAINING INSTEAD OF  
MXML PLAYING"



# Flex, the Professional Way

## Unique Flex Course by Renowned Author and Software Architect

While most books and other courses focus on teaching people how to use MXML, this course is unique in a way that it helps developers to build solid ActionScript foundation and to demonstrate the best practices of building application using Flex as the front-end and Java as the back-end.

As a renowned book author and software architect for large enterprise projects, Jack Li has been invited by many leading Fortune 500 companies to conduct this course. Their feedbacks show that it helps their employees gain essentials of ActionScript 3.0/Flex in few days and some employees are even able to start Flex projects right after the training.

Currently, Jack is still heavily involved in projects. However, he's willing to pull out three days every month to spread the joy of enterprise Flexing. If you need to equip yourself or your teams with real world Flex skills, do book early to avoid disappointment.

## 由軟件架構師兼知名IT作者執教的獨一無二Flex課程

大多數的書與課程都是在介紹Flex的表面功夫, MXML. 本課程的與眾不同之處在于它可以真正的幫助開發人員建立深厚的ActionScript基礎并且可以學習怎樣優雅的設計Flex前端Java后端的企業程序.

作為知名IT作家和軟件架構師, Jack Li廣受財富500強企業的邀請來實施這個課程. 他們的反饋顯示, 這個課程可以讓他們的雇員很快的掌握ActionScript 3.0/Flex的技術, 有些學員甚至可以在完成課程后便開始Flex的項目.

目前, Jack依然在領導很多項目. 忙碌之中, 他每月抽出三天與大家分享Flex的開發樂趣. 如果你或者你的團隊需要掌握真正的Flex技術, 敬請盡快預定, 以免向隅.

## Featured Clients



# Course Details

## Instruction Style

This course has two goals: 1) to equip trainees with solid ActionScript foundation; 2) to enable them to build RIA applications with Flex on the front-end and Java (or other languages) on the back-end right after the course.

In order to realize the two goals, the course is divided into two parts. The first part is AS3 foundation building and the second part is enterprise Flex with Java crafting. Take a three-day course as an example. The trainees will learn AS3 with a little touch on Flex framework on the first day. The following two days, they will code (yes, a lot of coding) Flex applications under the guide of the instructor. The best part is that the instructor will share his insights and thinking process with trainees so that they can use best practices and patterns to build the applications. Please visit <http://flex-training.googlecode.com> for sample course materials.

此課程有兩個主要目標：1) 幫學員樹立堅實的ActionScript基礎；2) 使學員可以很快的開始以Flex為前端Java為后端的RIA編程。為此，本課程分為兩部分，其一為AS基礎建設；其二為企業級Flex/Java編程。以三天的課程為例，學員第一天學習AS3及Flex的基礎；后兩天他們將在導師的幫助下進行Flex企業編程。最好的是，導師將分享他的Flex項目的實際經驗，學員可以很快的掌握最佳實踐。請訪問<http://flex-training.googlecode.com> 查看部分課程資料。

## Pricing

|            | Corporate Training 企業培訓                                                                                                                                                                                                                                              | Individuals 個人拼班                                                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| Pricing    | <b>RMB 25,000/Day</b><br>人民幣25,000每天                                                                                                                                                                                                                                 | <b>RMB 7,000/course</b><br>人民幣7,000每課程(三天)                                                             |
| Conditions | Conducted in English or Chinese. Clients provide the venue. 20 trainees. Price varies with number of trainees. The quote assumes the training location is at Shanghai; 20% surcharge applies for other cities or countries. 以英文或中文授課。客戶提供場地。20學員。價格因人數而變化。异地/國酌加20%。 | The training is provided monthly in Shanghai, CHINA or in SINGAPORE. Max 20. 培訓每月一期，地點為中國上海或新加坡。限制20人。 |
| Remarks    | Training can be customized according to the need of the trainees. Please provide trainees list and background information prior to the training.                                                                                                                     |                                                                                                        |

## How to book 預定

Tel: +86 15901631326 (Liang) Email: [flextraining@insprise.com](mailto:flextraining@insprise.com)

# The Instructor

## Sharing Wisdom with Global Professionals 著書立說 共享智慧

Since 1999, Jack has been writing numerous articles on a wide range of topics to share the wisdom of programming with peer developers. 從1999年開始，Jack寫了大量的文章與世界上的開發人員分享編程的奧妙，涉及相當多的方面。

developerWorks.



**JDNC simplifies Java UI development** *IBM DeveloperWorks, 27 Sep 2005.*

**Text-independent Speaker Verification Using Minimal Resource Allocation Networks**  
*International Journal of Neural Systems, Vol. 14, No. 6 (2004) 347-354*

**UI development with JavaServer Faces** *IBM DeveloperWorks, Sep 2003.*

**Build your stock with J2ME** *IBM DeveloperWorks, Oct 2002.*



**Professional Java Native Interfaces with SWT/JFace**

2005由Wrox在美國發行。售罄后，歷經重印。這是一本關於如何使用SWT/JFace進行UI編程的書。Published in 2005 by Wrox. Due to popular demand, this book has been reprinted.

## Solid Consulting Experience with World Clients 堅實經驗 環球客戶

Not believing in certifications, Jack has drawn much attention with his deep knowledge in the field. Since 2003, Jack has been invited as an independent consultant to provide consulting services to various government agencies, banks and consulting companies - EDS, IBM, StandardChartered Bank Corp, CA, Singapore ICA and Accenture. Jack provides them valuable insights to architecture design, algorithms, and patterns in their software projects. He is proud to serve the society directly with his knowledge. 軟件開發方面的精益求精，使Jack獲得頗多關注，不斷被政府，銀行以及各類諮詢公司聘為獨立的顧問，如EDS, IBM, CA, 埃森哲等。Jack為這些機構的軟件項目提供了寶貴的建議以及關鍵的架構設計和算法等。他以將知識直接為社會服務為榮。

## A Developer with Attitude 獨立人格 卓越追求

- Education Bachelor of Engineering & Master of Engineering, Nanyang Technological University, SINGAPORE 工程學士與碩士，南洋理工大學/新加坡  
Thermal Department, Tsinghua University, PEKING 清華大學/北京
- Honors Asia Mobile Java Challenge (AMJC) 2002/3, organized by SingTel, Nokia, and Sun Microsystems, Top 5 winner. The only individual developer short listed as a finalist and selected as Top 5. 新電信諾基亞升陽杯亞洲移動Java比賽，個人組第一名  
ACM International Collegiate Programming Contest (ICPC) 2001. Founder and leader of team C2L. 5th place in Asia, 2nd in Singapore. 2001 ACM ICPC - 亞洲第五名
- Citizenship Singaporean

# Thoughts on Flex Training

## The Instructor's Rich Experience on Flex

- Founder of Adobe User Group Shanghai (a.k.a. RIAShanghai.com) Adobe上海Flex User Group (a.k.a. RIAShanghai.com) 創始人
- Several articles published on Adobe websites Adobe網站上發表數篇文章
- Number of large Flex projects completed: 2; Line of code (Flex): > 100,000 領導大型Flex項目數: 2; 代碼行數(Flex only): > 十萬

## Philosophy of Flex Training

Jack believes that a language or a framework is of no use unless it can integrate and inter-operate with other languages and systems. He never talks about Flex without mentioning how it interacts with the Java server side and other systems. In fact, he has developed a Hibernate-like ROM framework that works transparently across the Java server side and the Flex/ActionScript client side. If you find your teams have no problem with Flex but have no idea about how to put it into real use, ask Jack. He'll show you best practices and patterns in the Flex context as well as the Java server side with solid support from his experience. "By far the best proof is experience." - Francis Bacon

Jack相信獨立的一門語言或者一個框架是沒有任何用處的，除非它可以與其它系統進行良好的交互。他從來都不只談論Flex，相反的，他會將所有相關的技術包含進來。實際上，他開發了一個類似于Hibernate的ROM框架，這個框架可以在Flex客戶端以及Java服務器端無縫的運行。如果你發現你的團隊掌握了Flex的技術，却無法在實際中去運用，那可以諮詢Jack。他會給你的團隊展示優雅的設計模式和推薦的編程方法，而這些都是基于他堅實的開發經驗。"By far the best proof is experience." – 培根

**Build your skills. Realize your value. Book now.**

Tel: +86 15901631326 (Liang) Email: flextraining@insprise.com