

Flex 3 でつくる はじめてのRIA

(リッチインターネットアプリケーション)

たいらひでかづ
アドビ認定インストラクター(ACI)



Who is this guy?



- たいらひでかづ
- m-Schoolインストラクター
- Flex および Flash アドビ認定インストラクター(ACI)



本日のテーマ



- Flex による RIA (リッチ・インターネット・アプリケーション) 開発の初歩をハンズオン (操作演習) を通して習得します。
- Flex の開発経験がない Web アプリケーション開発者を対象としています。代表的な UI コンポーネント (コンテナやコントロール) を配置したアプリケーション画面を作成し、Web サービスなどのリモートデータへのアクセスを行います。
- また、Flex で開発したアプリケーションを Adobe Integrated Runtime (AIR) 上で実行可能な形式に変更する方法も簡単にご紹介します。
- Flex Builder 3 (ベータ) を使用します。
- LiveCycle Data Services (LCDS) 2.5 Express Edition を使用します。



本日の内容



1. Flexの概要と基本シンタックス
2. MXMLカスタムコンポーネント
3. コンポーネントの使用に関して
4. Webサービス
5. イベント処理
6. リモート Java オブジェクトへのアクセス (LCDS登場)
7. DataGrid の概要
8. Web版のFlexアプリケーションをAIR版に書き換える



セッションデータの公開

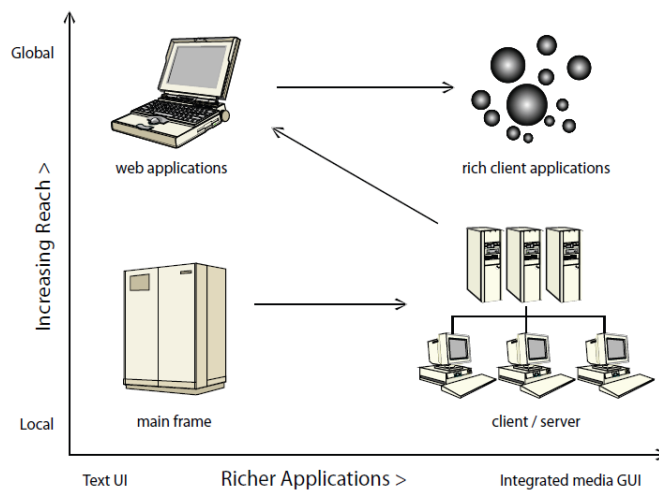


- 以下の URL で 11月5日(月)以降、**順次**公開させていただきます
http://m-school.biz/event/2007_max_japan.htm

- 著作権の関係上、公開できないファイルがございます。



P.5 リッチ・インターネット・アプリケーションの理解



P.6 Flexの製品群



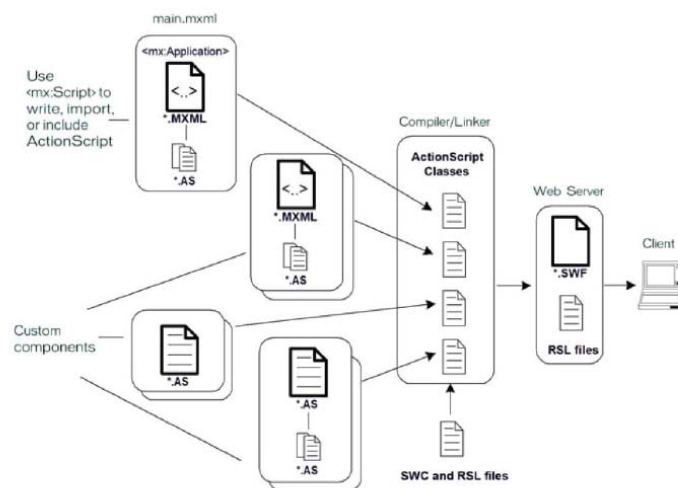
- Adobe Flex 3 SDK
 - 無償
- Adobe Flex Builder 3
 - 開発環境
- Charting
 - チャートコンポーネント
- LiveCycle Data Services (旧・Flex Data Services)
 - Flexアプリケーションにおけるメッセージインフラストラクチャ(データ通信の土台)

2007 Adobe Systems Incorporated. All Rights Reserved.

7



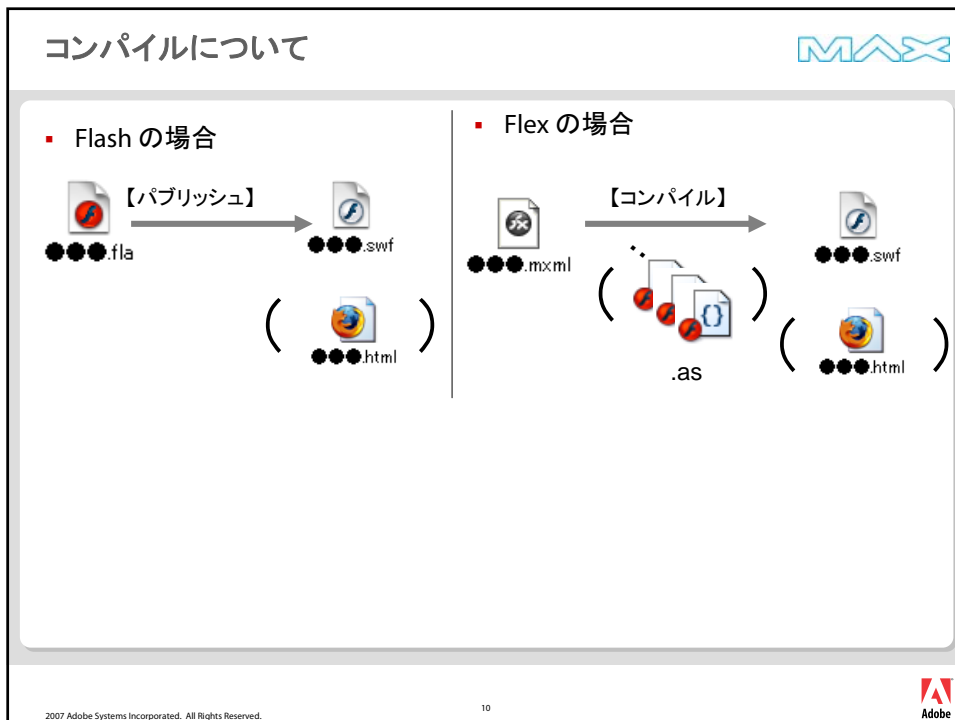
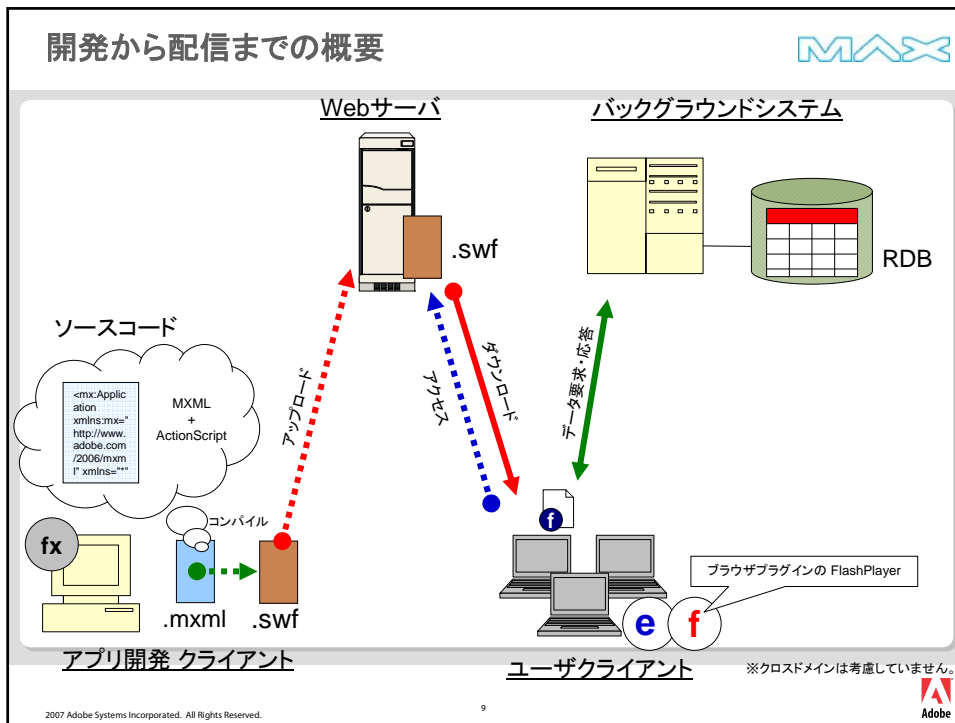
P.7 開発から配信までの流れ



2007 Adobe Systems Incorporated. All Rights Reserved.

8





P.8 MXMLの基本シンタックス



```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute">
  <!-- body of application here -->
</mx:Application>
```



P.10 インスタンス名



```
<mx:Label id="labelOne" text="Hello"/>
```



P.10 プロパティ値設定の2つの方法



- ▶ Component properties can be specified in two ways

- ▶ As attributes

```
<mx:Label text="Hello"/>
```

- ▶ As nested tags (also called "nodes")

```
<mx:Label>  
  <mx:text>Hello</mx:text>  
</mx:Label>
```

- Property tag names always begin with a lower-case name
- Component tag names are always upper-case



P.10 バインディング



- バインディング

- あるインスタンスのプロパティを「動的に」ほかのインスタンスのプロパティに反映させる。

```
<mx:Label id="labelOne" text="hello"/>  
<mx:Label text="{labelOne.text}"/>
```



P.13 UIコンポーネント



- コントロール
 - 要素・部品
 - Button、TextInput、List など
- コンテナ
 - レイアウトを担当
 - さらに2つに分かれる
 - レイアウトコンテナ
 - 「静的な」レイアウト担当
 - VBox、Hbox、Canvas など
 - ナビゲータコンテナ
 - 画面遷移担当
 - ViewStack、Accordion、TabNavigator



P.13 layoutプロパティとレイアウトコンテナの対応



- *vertical* - all components stack vertically **VBox**
- *horizontal* - all components line up horizontally **HBox**
- *canvas* - each component must be assigned a specific x and y position **Canvas**



P.17 MXMLコンポーネント側



```
<?xml version="1.0" encoding="utf-8"?>
<mx:VBox
  xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:Image source="assets/header.jpg"
    x="137" y="25"/>
  <mx:Label text="Image Info"/>
</mx:VBox>
```



P.17 メインアプリケーション



- <mx:Application>があるファイルを「メインアプリケーション」ということがある。

```
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  xmlns:comp="components.*" >

  <comp:Component1 />

</mx:Application>
```



P.18 [Bindabel] メタキーワードが必要な場合



- 宣言した変数(プロパティ)をバインディングする場合
- コンパイル時
 - 警告表示にはなるが、コンパイルエラーとはならない。
- 動作上の問題
 - 変数(プロパティ)の値の変化が検出されない。つまり、値が動的に反映されない。



P.23 WebService



1. インスタンス化とアクセス先の指定
 - `<mx:WebService id="【インスタンス名】" wsdl="【WSDLを指定】">`
2. 送受信のトリガ
 - `【インスタンス名】.【呼び出すメソッド名()】`
 - ユーザイベントで呼び出す
 - `<mx:WebService>`自身のloadイベントで呼び出す
 - 「WSDLの読み込み完了時」
 - `<mx:Application>`のcreationCompleteイベントで呼び出す
 - 「Flexアプリケーションの初期化終了時」
3. 受信
 - a. 受信したデータを単純に表示
 - `【インスタンス名】.【呼び出したメソッド名】.lastResult`
 - b. 受信したデータを元に処理をする
 - `<mx:WebService>`のresultイベントでActionScriptの処理関数を呼び出す。



P.26 イベント



■ イベントの種類

- ユーザイベント
 - ユーザによるマウスイベント、キーボードイベント
- システムイベント
 - アプリケーション実行中に自動的に発生するイベント

■ イベントオブジェクト

- イベントが発生すると必ず生成されるオブジェクト。
- flash.events.Event クラス(またはそのサブクラス)。
- eventという文字列で、ActionScriptの関数の引数に渡せる。
- イベントオブジェクトには、役立つ情報が満載。

```
<mx:Button label="Click Me" id="button1"
click="fillLabelControl(event)"/>
```



P.33 RemoteObject



1. インスタンス化とアクセス先の指定

- <mx:RemoteObject id="【インスタンス名】" wsdl="【LCDS上のJavaクラスを指定】">
- LCDS上のJavaクラス
 - POJO(Plain Old Java Object)を作成し、WEB-INF\classesフォルダに配置。
 - WEB-INF\flex\remoting-config.xml を設定。

2. 送受信のトリガ

- 【インスタンス名】.【呼び出すメソッド名()】
- ユーザイベントで呼び出す
- <mx:Application>のcreationCompleteイベントで呼び出す

3. 受信

- a. 受信したデータを単純に表示
 - ・ 【インスタンス名】.【呼び出したメソッド名】.lastResult
- b. 受信したデータを元に処理をする
 - <mx:RemoteObject> のresultイベントでActionScriptの処理関数を呼び出す。



```
<mx:RemoteObject id="catR0"
  destination="sandwichService">

  <mx:method name="getTypes"
    result="catHandler(event)"/>

  <mx:method name="getSandwichesByType"
    result="sandwichHandler(event)"/>

</mx:RemoteObject>
```

ありがとうございました。

もっと習得したい方はこちら。

Fxのトレーニングなら **m-School**
<http://m-school.biz>

