

「Writing multiscreen AIR apps」の日本語訳

本ドキュメントは、insideria.com サイトで公開されている記事「Create your 1st Flex based AIR on Android Application」をヒム・カンパニー 永井勝則が自主的に日本語に訳したものです。

<http://www.himco.jp/>

[knagai@himco.jp](mailto:knagai@himco.jp)

(2010/11)

本ドキュメントの原文は

<http://insideria.com/2010/10/create-your-1st-flex-based-air.html>

です。

## 初めての Flex ベースの AIR Android アプリケーション

Rich Tretola

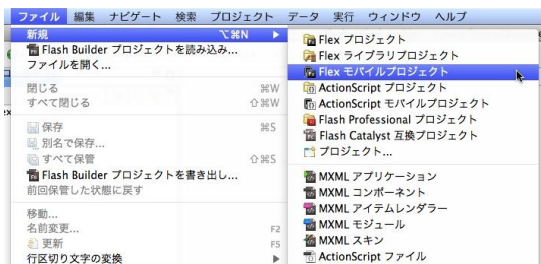
Christian Cantrell

2010/10/25

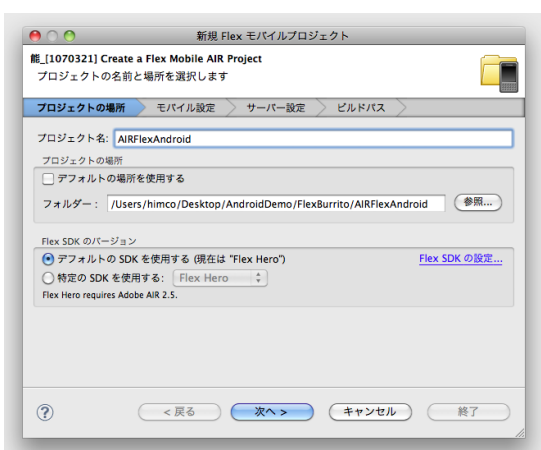
ついに Flex Builder Burrito が登場し、みなさんとはにかくにも Flex ベースの Android アプリケーションの作成を始めたいと思っておられることでしょう。このチュートリアルでは、Android で動作する Flex ベースの Adobe AIR マルチスクリーンアプリケーションの作成方法を見ていきます。

[Flash Builder パッケージとソースのダウンロード: AIRFlexonAndroid.fxp](#)

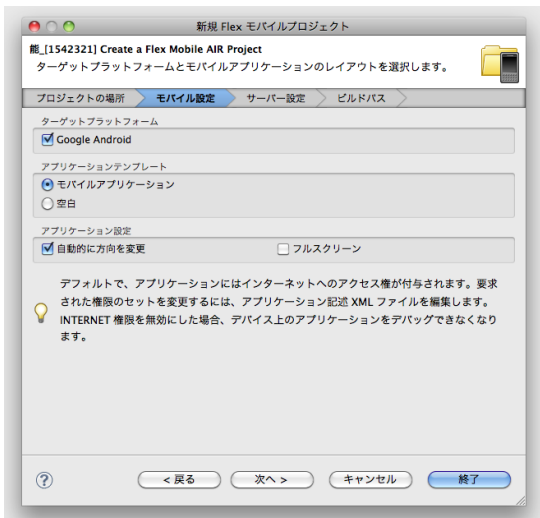
手順1: [ファイル]→[新規]→[Flex モバイルプロジェクト]を選択します。



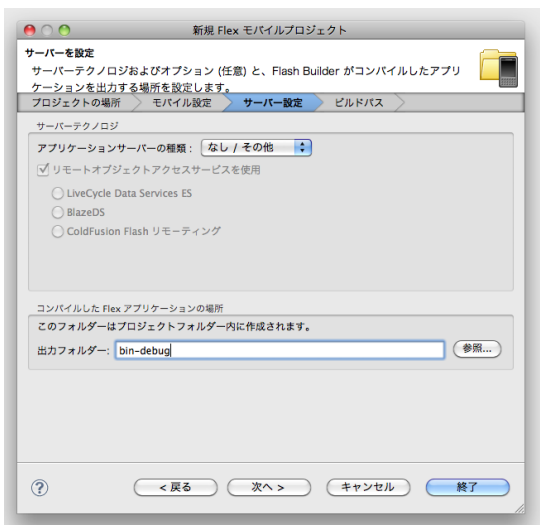
手順2: [プロジェクト名]に AIRFlexonAndroid と入力します。



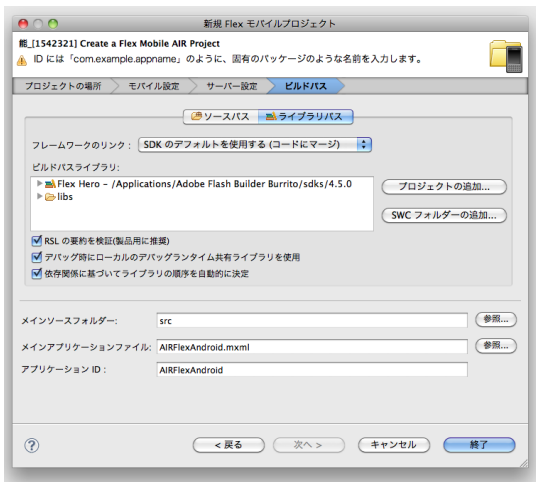
手順3: [モバイル設定]では[Google Android]と[モバイルアプリケーション]が選択されていることを確認します。



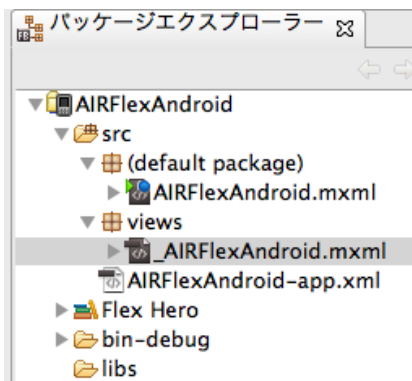
手順4: アプリケーションサーバーは使用しないので[サーバー設定]では何も変更しません。



手順5: 何も変更せず、[終了]をクリックします。



Flex モバイルプロジェクトが新たに作成されます。デフォルトファイル(AIRFlexonAndroid.xml)と初期ビューのファイル(vIEWS パッケージの\_AIRFlexonAndroid.xml)が自動的に作成される点に注目してください。



プロジェクトが作成できたので、コードを追加していきましょう。この簡単なデモでは、ナビゲーションシステムと3つのスクリーンを作成します。

ホームナビゲーションの作成:

AIRFlexonAndroid.xml という名前をついたデフォルトファイルを開き、<fx:Declaration>の下に、以下のコードを追加します。これは、クリックされたらビルトインのナビゲータ(navigator)に popToFirstView することを伝える Button です。FirstViewとは何ぞや?と思われるでしょう。Android の Flex AIR は ViewNavigator にもとづいており、ビューを変更するにはただナビゲータにやることを伝えるだけで、後はナビゲーションシステムがそれに応答してくれるのです。またタグ内では、firstView プロパティが views.\_AIRFlexonAndroid に設定されていることにも気づかれるでしょう。これはプロジェクトを作成したときに自動的に生成されたファイル(\_AIRFlexonAndroid.xml)です。ではこの Home ボタンをクリックすると、どうなると思いますか? その通り、views.\_AIRFlexonAndroid ビューがロードされるので

す。

```
<s:navigationContent>
  <s:Button label="Home" click="navigator.popToFirstView()"/>
</s:navigationContent>
```

ここまでの AIRFlexonAndroid.mxml ファイルは次のようになっています。

```
<?xml version="1.0" encoding="utf-8"?>
<s:MobileApplication xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark" firstView="views._AIRFlexAndroid">
  <fx:Declarations>
    <!-- 非ビジュアルエレメント（サービス、値オブジェクトなど）をここに配置 -->
  </fx:Declarations>
  <s:navigationContent>
    <s:Button label="Home" click="navigator.popToFirstView()"/>
  </s:navigationContent>
</s:MobileApplication>
```

#### スクリーンの追加

次は、\_AIRFlexonAndroid.mxml ファイルに内容を追加していきましょう。views パッケージからこのファイルを開き、</fx:Declarations>の後に次のコードを記述します。

```
<s:Button label="Go to Screen 1"
  click="navigator.pushView(views.Screen1)"
  horizontalCenter="0"
  verticalCenter="0"/>
```

\_AIRFlexonAndroid.mxml 全体は次のようになっています。

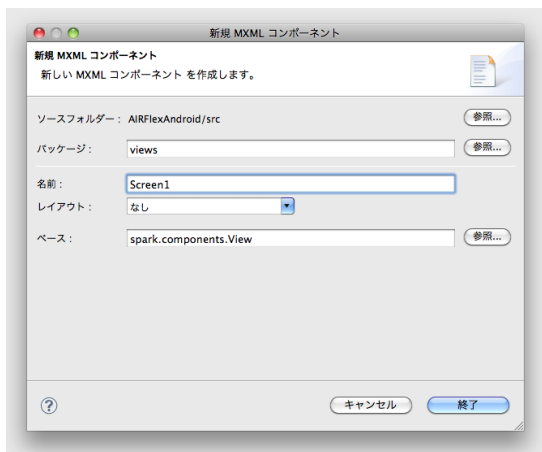
```
<?xml version="1.0" encoding="utf-8"?>
<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark" title="ホーム">
  <fx:Declarations>
    <!-- 非ビジュアルエレメント（サービス、値オブジェクトなど）をここに配置 -->
```

```

</fx:Declarations>
<s:Button label="Go to Screen 1"
           click="navigator.pushView(views.Screen1)"
           horizontalCenter="0"
           verticalCenter="0"/>
</s:View>

```

\_AIRFlexonAndroid.mxml ファイルには、navigator.pushView(views.Screen1)というクリックアクションを持つ Button を加えました。これはナビゲータにビューを views.Screen1 に変えるよう伝えます。しかしわれわれはまだこのファイルを作成していません。この新しい View を作成するには、views パッケージを右クリックし、[新規]→[MXML コンポーネント]を選択します。[名前]に Screen1 を指定して[終了]をクリックします。



Screen1 を以下のコードで更新します。ここではボタンを2つ作成しています。1つは navigator.popToFirstView() を使うことでホームに戻るボタンです。もう1つは navigator.pushView(views.Screen2)関数を呼び出すことでスクリーン2に進むボタンです。また title プロパティが Screen1 に設定されていることにも注意してください。これはナビゲーションに重要です。

```

<?xml version="1.0" encoding="utf-8"?>
<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"
        xmlns:s="library://ns.adobe.com/flex/spark" title="Screen1">
    <fx:Declarations>
        <!-- 非ビジュアルエレメント (サービス、値オブジェクトなど) をここに配置 -->
    </fx:Declarations>

```

```
<s:VGroup horizontalCenter="0" verticalCenter="0">
    <s:Label text="I am screen1"/>
    <s:Button label="Go Home"
                click="navigator.popToFirstView()"/>

    <s:Button label="Go To Screen 2"
                click="navigator.pushView(views.Screen2)"/>

</s:VGroup>
</s:View>
```

同じ作業を繰り返し、Screen2.mxml を作成して以下のコードに更新します。Screen2 にもボタンが2つあり、その title は Screen2 です。

```
<?xml version="1.0" encoding="utf-8"?>
<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"
        xmlns:s="library://ns.adobe.com/flex/spark" title="Screen2">
    <fx:Declarations>
        <!-- 非ビジュアルエレメント（サービス、値オブジェクトなど）をここに配置 -->
    </fx:Declarations>

    <s:VGroup horizontalCenter="0" verticalCenter="0">
        <s:Label text="I am screen2"/>
        <s:Button label="Go Home"
                    click="navigator.popToFirstView()"/>

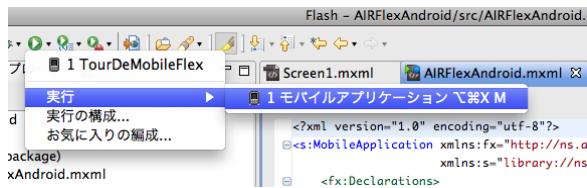
        <s:Button label="Go To Screen 1"
                    click="navigator.pushView(views.Screen1)"/>

    </s:VGroup>

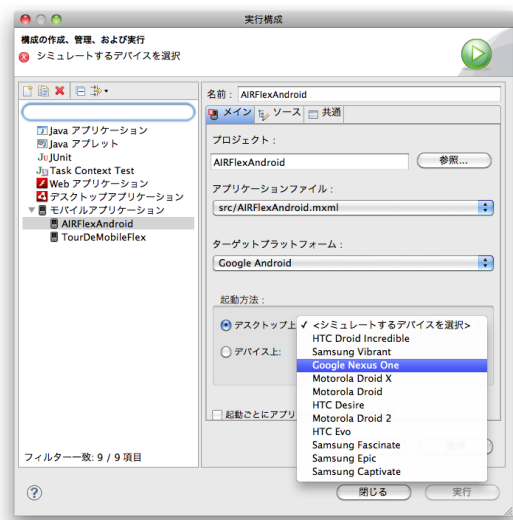
</s:View>
```

#### エミュレータでの実行

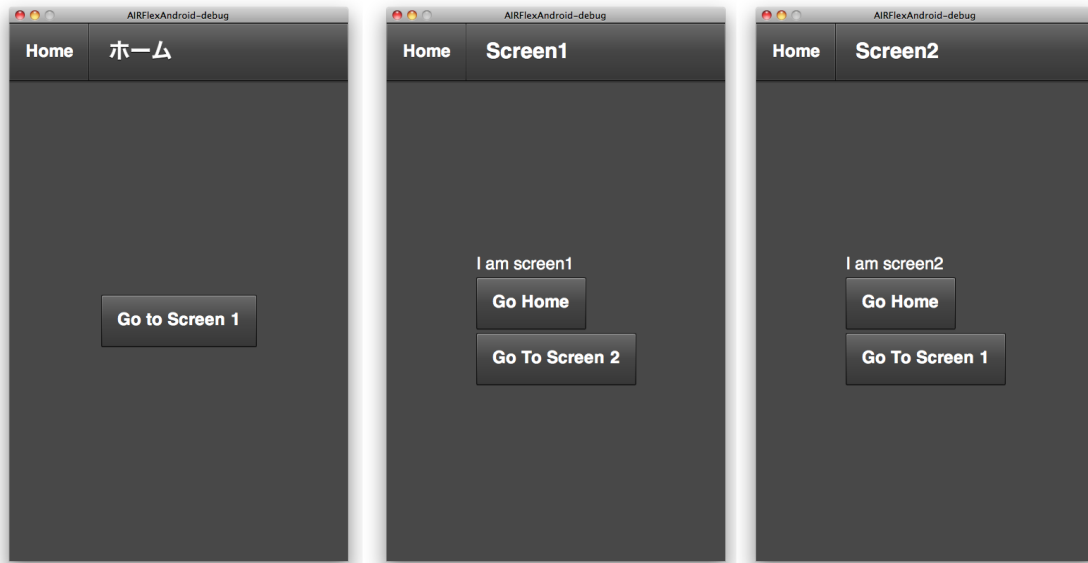
ここまでの作業で、アプリケーションがエミュレータ内で実行できるようになります。AIRFlexonAndroid.mxml ファイルを開き、実行ボタンの右の三角マークをクリックして[実行]→[モバイルアプリケーション]を選択します。



すると下図のウィンドウが開きます。[起動方法]で[デスクトップ上]を選択し、シミュレーションするデバイスを選びます。下図では Google Nexus One を選んでいます。[適用]をクリックし[実行]をクリックします。



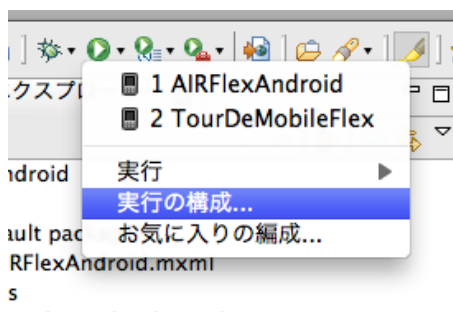
エミュレータが開きます。エミュレータではアプリケーションのボタンや上部の[Home]ボタンを使って、ビューを行き来できます。[Home]ボタンの右にはスクリーン名 (\_AIRFflexonAndroid.mxml の title="ホーム" 設定により)が表示されています。これはすでに述べたように、View の title プロパティで変更することができます。



以上がエミュレータでの実行です。

#### デバイス上での実行

Android デバイスをお持ちのみなさんはこのチュートリアルをさらに進めることができます。デバイスを USB ポートに接続し、[実行]→[実行の構成]を選択します。



今度は[起動方法]を[デバイス上]に変更し、[適用]をクリックして[実行]をクリックします。AIR アプリケーションがデバイスにインストールされ、起動します。

