

Introduction to Flex

CS174

Chris Pollett

Nov. 26, 2007.

Outline

- Flash and Flex
- Installation
- Simple Program
- Components

Flash

- Flash was originally a product to make it easy to produce vector based animations on the web created around 1996 based on technology purchased by Macromedia (now Adobe) from Futurewave.
- Flash programs are typically executed by the Flash player plug-in which currently is (June 2007) available in some browser on 99% of desktops.
- Nowadays, it is used to create Rich Internet Applications (RIA) and competes against AJAX, Silverlight (Windows), Applets (Java), etc.
- It can also be used to do tracking (PIE -- Persistent Identification Elements -- not turned off as often as cookies)
- It has the advantage over AJAX that as the plug-in is produced by one source, it runs the same in any browser. It also supports richer kinds of media like Video. (YouTube).

Flex

- The main authoring tool for Flash is Flash Professional, which is GUI IDE designed to make it easy to create Flash.
- Flex is an Adobe product which gives a more programmatic way to create Flash files.
- There is a Flex IDE, Flex Builder, which can also be used, however, the basic SDK is open sourced under the Mozilla License (can mix proprietary and open source code).
- We will only look at the basic SDK which we will access mainly from the command line.
- There are other open-source tool chains which can be used to create .swf files, notably, OpenLaszlo.

Installation

- Currently, the release version of Flex is Flex 2, although Flex 3 beta does exist.
- It can be found at:
<http://labs.adobe.com/technologies/flex/sdk/>
- Download this file and unzip it somewhere you know, say C:\FlexSDK (Windows) or /usr/local/FlexSDK (*nix including Mac)
- Then add C:\FlexSDK\bin or /usr/local/FlexSDK/bin to your PATH variable (use control panel Windows, or edit .profile, .cshrc, etc on *nix).
- Restart your shell.

Creating a simple program

- Creating a Flex App involves writing a mxml file which is a XML language which may be scripted (as Javascript can be used to script HTML) using Adobe's Actionscript language (a variant of Javascript).
- You then compile this code using the mxmmlc compiler.
- To include the compiled code into HTML file you need to use an object/embed tag

A simple MXML file (test.mxml)

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  width="100%" height="100%" >
  <mx:Button label="Hello World
  Button"></mx:Button>
</mx:Application>
```

This could then be compiled by typing:

```
mxmlc test.mxml
```

This should produce a test.swf file.

Including your .swf file in a web page

- To include your program in a web page you need to an object/embed tag.
- For instance, you could use:

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  id="test" width="100%" height="100%"
  codebase="http://fpdownload.macromedia.com/get/flashplayer/current/sw
  flash.cab">
  <param name="src" value="test.swf" />
  <embed
    src="test.swf"
    name="test"
    width="100%" height="100%"
    pluginspage="http://www.adobe.com/go/getflashplayer">
  </embed>
</object>
```


Components

- Flex applications are built out of components.
- There are visual components such as Containers and user interface controls
- There are non-visual components such as data components and utility components.
- Let's take a look at these.

Containers and UI controls

- A container is a component that can be used to hold other components and might be used for layout purposes:

```
<mx:VBox>
```

```
  <mx:Button label="Button 1" />
```

```
  <mx:Button label="Button 2" />
```

```
</mx:VBox>
```

There is also an HBox tag, tags for gridlayout, tiles, etc.

Layouts can be nested.

- UI Controls. We already seen Button. There are many others such as ComboBox, Text, TextInput, etc.
- Properties of a Control can be set using either: tag attributes, nested tags, or ActionScript.
- You often give tags id attributes. The UI control object can then be used elsewhere as a variable (data binding) provided the id name is is put in {}.
- See next slide for an examples.

Example 2

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" width="100%"
  height="100%" >
  <mx:Vbox>
    <mx:HBox>
      <mx:Button> <mx:label>Button 1</mx:label></mx:Button>
      <mx:Button label="Button 2" />
    </mx:HBox>
    <mx:ComboBox>
      <mx:dataProvider>
        <mx:ArrayCollection>
          <mx:String>Item 1</mx:String>
          <mx:String>Item 2</mx:String>
          <mx:String>Item 3</mx:String>
        </mx:ArrayCollection>
      </mx:dataProvider>
    </mx:ComboBox>
    <mx:Text text="hello" width="100" height="15" />
    <mx:Text text="{input.text}" width="150" height="20" />
    <mx:TextInput id="input" />
  </mx:VBox>
</mx:Application>
```