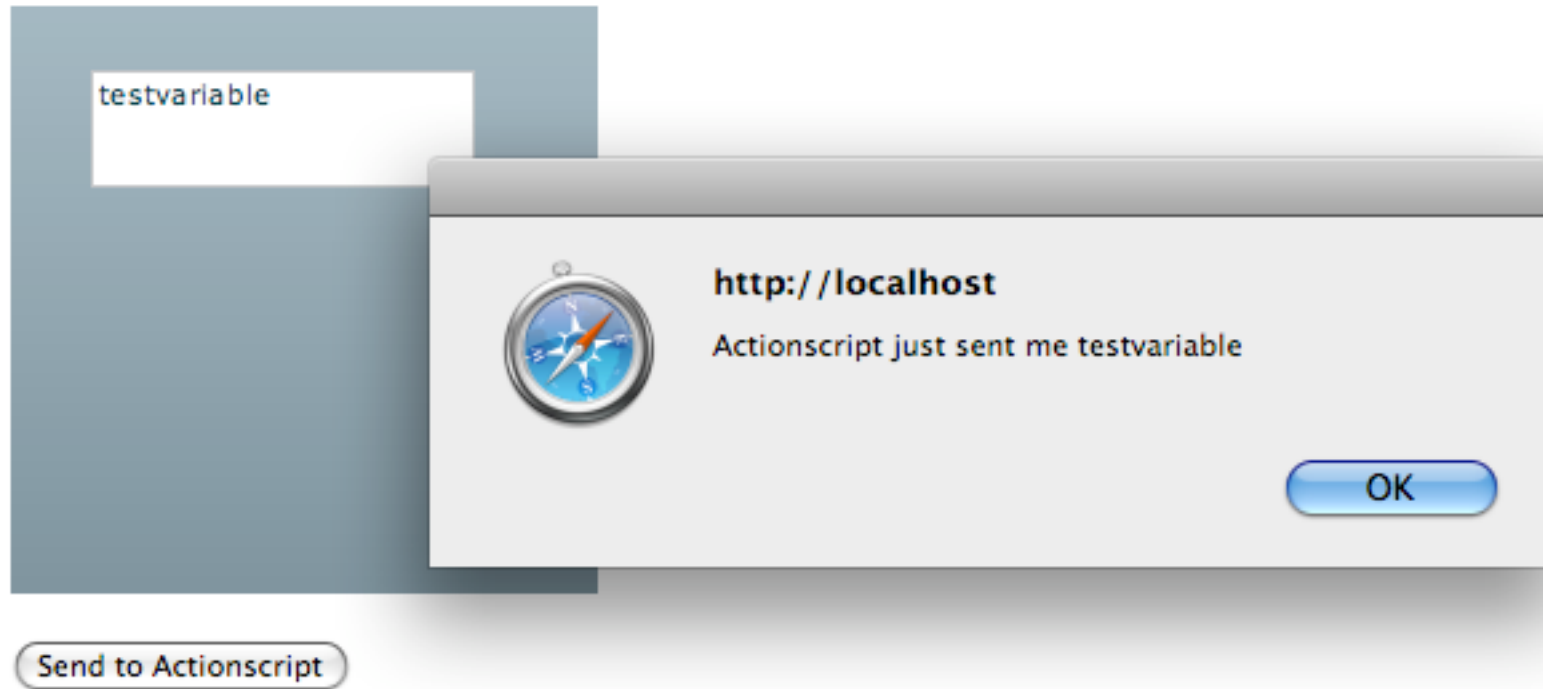# Even More Flex

CS174

Chris Pollett

Dec 3, 2007.

Outline

- HTML ActionScript data exchange
- Extended Sound Example

Data Exchange

- It is sometimes useful for HTML and Flex/Actionscript to be able to communicate with each other.

- For example, you might want to use the same .swf but send in the start up parameters different videos to play.

- Or, as people are more familiar with HTML UI's, you might have HTML buttons to control things in Flex or vice-versa.

- We'll look at how to do both of these things in preparation for a more complicated media example involving playing MP3's.

Data Exchange -- Screenshot

testvariable

http://localhost

Actionscript just sent me testvariable

OK

Send to Actionscript

Data Exchange Test -- HTML

```html
<html>
<head>
<script type="text/javascript">
    //this calls an Actionscript function
   function sendToActionScript(value)
   {
      document.getElementById('testParam').sendToActionScript(value);
   }
   // this is called from Actionscript
   function receivedFromActionScript(value)
   {
      alert("Actionscript just sent me "+value);
   }
</script>
</head>
<body>
<object id="testParam" type="application/x-shockwave-flash" data="testParam.swf"
width="220" height="220">
<param name="src" value="testParam.swf" />
<!-- flashvars used to send data into Actionscript. Can also use query string -->
<param id="flashvars" name="flashvars" value="testvar=testvariable" />
<param name="allowScriptAccess" value="always" /><!-- for Liveconnect -->
</object>
<p><button onclick=" sendToActionScript('hi');">Send to Actionscript</button></p>
</body>
</html>
```

Data Exchange Test -- Flex

```
<?xml version="1.0" encoding="utf-8"?> //notice we handle preinitialize and activate events
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"  width="100%" height="100%"
    preinitialize="init();" activate="sendToJavascript(my_text.text);">
 <mx:VBox>
 <mx:Script>
 <![CDATA[
[Bindable] private var testvar:String; //Bindable makes available to Flex inline expressions
private function init():void //get param data sent, set up receivedFromJavascript call when Jscript calls
{
   testvar=Application.application.parameters.testvar;
   flash.external.ExternalInterface.addCallback("sendToActionScript", receivedFromJavascript);
}
private function receivedFromJavascript(value:String):void
{
    my_text.text= value; //changes TextArea contents below
}
private function sendToJavascript(value:String):void //used to call Javascript function when App focussed
{
   flash.external.ExternalInterface.call("receivedFromActionScript", value);

    my_text.text= value;
} ]]>
</mx:Script>
<mx:TextArea id="my_text" width="90%" height="90%" text="{testvar}" />
</mx:VBox>
</mx:Application>
```

Sound

- We're now going to give an example of how to write an MP3 player using Flash.
- This is slightly more complicated than the Video example we did last day, as the Flex tag for sound -- SoundEffect -- is quite limited so we use the Flash Sound object instead.
- This example illustrates several things:
  - Having a separate .as files for our Actionscript classes.
  - Using information passing HTML -- Actionscript
  - Using [Bindable] with Actionscript classes/objects
  - How to write set and get methods in Actionscript
  - Sound, SoundChannel classes
  - Handling several new kinds of events and the EventDispatcher class: TimerEvent, ID3Event, change, valueAccept
  - More use of the Canvas tag to superimpose controls
  - New UI Controls : Spacer, ProgressBar

Sound Example -- HTML

```html
<html>
<head><!-- this is the HTML file used with our Flex App. It also scripts which song to play-->
<script type="text/javascript">
    function sendMp3(mp3Path)
    {
       mp3player =document.getElementById('testMp3');
       mp3player.sendMp3(mp3Path);
    }
</script>
</head>
<body>
<p style="margin-left:120px;">Play song:
<select onchange="sendMp3(this.options[this.selectedIndex].value)">
  <option>testA.mp3</option>
  <option>testB.mp3</option>
</select>
</p>
<object id="testMp3" type="application/x-shockwave-flash" data="testMp3.swf" width="400"
    height="200">
<param name="src" value="testMp3.swf" />
<param name="allowScriptAccess" value="always" />
</object>
</body>
</html>
```

Sound Example -- Actionscript I

```actionscript
package
{  // MP3 player below is the main class used to play an MP3
    import flash.events.*;
    import flash.media.*;
    import flash.net.*;
    import flash.utils.*;
    import mx.controls.*;
    public class MP3Player extends EventDispatcher //will use this classes dispatchEvent method
    {
      [Bindable] public var songName:String = "No Song Name Loaded"; //to bind field externally
      [Bindable] public var  time:String  = "0:00"; // need to bind within class, and then again object
      [Bindable] public var  progress:Number  = 0;   // of class in Flex
       private var _song : SoundChannel;
       private var _sound : Sound;
       private var _position : Number;
       private var _positionTimer:Timer;
       public function MP3Player( songURL : String )
       {
            _sound = new Sound( new URLRequest(songURL) );
            _sound.addEventListener(Event.ID3, id3Handler);
            _positionTimer = new Timer(1000); //goes off every 1000 msec to update ProgressBar
            _positionTimer.addEventListener(TimerEvent.TIMER, positionTimerHandler);
            _positionTimer.start();
            _position = 0; // playhead position msec.
            play();
       }
```

Sound Example -- Actionscript II

```
public function play():void // plays the mp3
{
    if(_song != null){ _song.stop(); }
    _song = _sound.play(_position);
}
public function formattedTime():String //pretty print duration of song
{
    var tsec:int = ((int)(_sound.length/1000)) % 60;
    var tmin:int = ((int)(_sound.length/60000)) % 60;
    var strsec:String = (tsec <10) ? "0"+tsec : ""+tsec;
    var timeMinSec:String = "" + tmin + ":" + strsec;
    return timeMinSec;
}
public function movePlay(val:Number):void //start playing at a different location
{
    if(_song != null) {_song.stop();}
    _song=_sound.play(val*_sound.length);
}
public function get length():Number //length of mp3 in msec
{
    return _sound.length;
}
```

Sound Example -- Actionscript III

```
public function get volume():Number //notice how Actionscript creates setter's and getter's
{
    return _song.soundTransform.volume;
}
public function set volume( vol : Number):void
{
    var transform:SoundTransform = _song.soundTransform;
    transform.volume = vol;
    _song.soundTransform = transform;
}
public function pause():void
{
    _position = _song.position;
    _song.stop();
}
public function stop():void  {        _song.stop();}
public function timerStop():void  {        _positionTimer.stop();}
private function id3Handler(event:Event):void //id3 are text tags at the end of an mp3 file which
{   songName = _sound.id3.songName;  // contain info about it. Whole file needs to load to find them
    time= formattedTime(); //That's why this handler might not get called immediately
    dispatchEvent(new Event("this could be anything")); // this will update binded variables
}
private function positionTimerHandler(event:Event):void
{ progress = _song.position/_sound.length; }
    }
}
```

Sound Example -- Flex I

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"   width="400" height="200"
      initialize="init();">
 <mx:Script>
 <![CDATA[ //notice we bind our player, this allows the binded fields it contains to work
[Bindable]
public var mp3Player:MP3Player  =  new MP3Player("testA.mp3"); //set up out player
private function init():void // set up Jscript-Ascript interface
{
   flash.external.ExternalInterface.addCallback("sendMp3", receiveMp3);
}

private function receiveMp3(songURL:String):void //used to switch song
{
   if(mp3Player != null)
   {
      mp3Player.timerStop();

      mp3Player.stop();
   }
   mp3Player = new MP3Player(songURL);
}


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

Sound Example -- Flex II

```
<mx:VBox>
   <mx:HBox>
      <mx:Spacer width="80" /><!-- moves us over 80 pixels -->
      <mx:Text id="songText" text="{ 'Song:'+mp3Player.songName}" />
   </mx:HBox>
   <mx:ControlBar>
      <mx:Text  text="Time: 0:00"  />
      <mx:Canvas>
         <mx:ProgressBar  id="progbar" x="0" y="21" maximum="1" width="80"  mode="manual"
   label="" />
         <mx:HSlider id="timeControl"  x="0" y="12" maximum="1" width="80"  value="{
   mp3Player.progress}" ><!-- called if user changes slider-->
            <mx:change>mp3Player.movePlay(timeControl.value);</mx:change>
            <!--called if slider value changes -->
             <mx:valueCommit>progbar.setProgress( mp3Player.progress +.05,1)</mx:valueCommit>
         </mx:HSlider>
      </mx:Canvas>
      <mx:Text  id="totalTime" text="{mp3Player.time}"  />
       <mx:Text  text="Vol:"  />
         <mx:HSlider id="volumeControl"  maximum="1" width="80" >
            <mx:initialize>volumeControl.value = .7; mp3Player.volume = volumeControl.value;
   </mx:initialize>
            <mx:change>mp3Player.volume = volumeControl.value; </mx:change>
         </mx:HSlider>
   </mx:ControlBar>
```

Sound Example -- Flex III

```
<mx:HBox>
    <mx:Spacer width="60"  />
  <mx:ControlBar>
    <mx:Button label="Play" click="if(mp3Player != null) mp3Player.play();" />
    <mx:Button label="Pause" click="if(mp3Player != null) mp3Player.pause();" />
    <mx:Button label="Stop" click="if(mp3Player !=null) mp3Player.stop();" />
   </mx:ControlBar>
   </mx:HBox>
   </mx:VBox>
</mx:Application>
```

States