



There is something in the AIR

Fasten your seat-belts !

Flight instructor : Koen De Weggheleire - www.newmovieclip.com



Flight contents

- ✓ Installing the AIR runtime
- ✓ Creating first AIR application
 - ➔ Case study
- ✓ Understanding AIR file structure
- ✓ Working with native Windows
 - ➔ Case study
- ✓ Working with the file system
 - ➔ Case study



Installing the AIR runtime

Instructions before taking off !

Flight instructor : Koen De Weggheleire - www.newmovieclip.com



Installing the runtime

- ✓ Download AIR runtime at
 - ➔ <http://labs.adobe.com/downloads/air.html>
 - ➔ runtime must be installed on each computer that will run AIR applications
- ✓ Ones installed: running AIR applications is like running any other desktop application.
 - ➔ Windows: double click application icon or select from start menu
 - ➔ Mac OS: double click application/ default installed in applications directory.

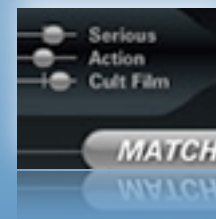
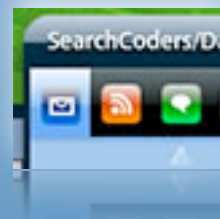
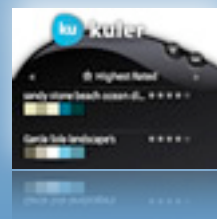
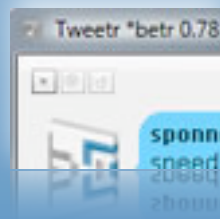
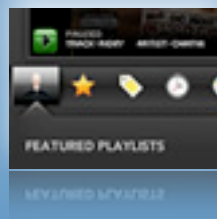


Adobe AIR and Flex 3

- ✓ Flex builder 3 available on
➔ labs.adobe.com

ADOBE FLEX™ 3

- ✓ Many sample applications on labs site to see what is possible:





Put your hands up in the AIR

And say HELLO AIR !

Flight instructor : Koen De Weggheleire - www.newmovieclip.com



Creating first AIR application

- ✓ Open Flex Builder 3
- ✓ Select file > new > Air project
- ✓ Accept default project location and Flex SDK version and click Next.
- ✓ Make no changes in build paths. Click Next.
- ✓ Specify setting for Air application
 - ▶ **ID** : HelloAir, **Name**: HelloAir
 - ▶ **Description**: First AIR app, **Copyright**: 2007



Creating first AIR application

- ➔ Add title attribute to WindowedApplication component.
- ➔ Add label component to the application and set text property of the label.
- ➔ Set horizontalCenter and verticalCenter to 0.
- ➔ Run application
- ➔ Resize window to see how label remains centered.



Export your first application

- ✓ Select File > Export
- ✓ Select AIR package and Click Next
- ✓ Click finish button to create Air package
- ✓ Install application by double-clicking it.



Understanding AIR file structure

- ✓ Two files are required for AIR applications
 - ➔ .AIR file : used to package AIR application
 - ➔ application.xml: defines various properties of the application.
- ✓ AIR files are actually ZIP files:
 - ➔ rename your application .AIR file to .ZIP and open your application xml file.
 - ▶ AIR XML namespace, appId attribute, version attribute, Application properties,...



Understanding AIR file structure

- ➔ in the root content node, set systemChrome and transparent attributes.
 - ▶ systemChrome="standard"
 - ▶ transparent="false"
- ➔ Run the application again.



Look through the window

there is something native in the AIR

Flight instructor : Koen De Weggheleire - www.newmovieclip.com



Working with Native Windows

- ✓ Using the AIR windowing API to create and manipulate windows.
- ✓ Using the `NativeWindow` class
 - ➔ A reference to a `NativeWindow` instance can be accessed using the `Stage.window` property
 - ➔ Adding Flex components directly to the display list of an application-created window is not supported in the Beta release of AIR
 - ➔ But you can load a Flex-based SWF into a new window (same domain)



- ✓ Using the NativeWindowInitOptions class:
 - ➔ specifies the initialization options for constructing new NativeWindows (maximizable, transparent, resizable, systemChrome, hasMenu...)
- ✓ Adding content to Native Windows (custom buttons, resizeHandlers,...)
 - ➔ Add to stage displayList of the Native Window



Working with Native Windows

- ✓ Add custom buttons to a chromeless window:
 - ➔ `<nativewindow>.stage.window.minimize();`
 - ➔ `<nativewindow>.stage.window.maximize();`
 - ➔ `<nativewindow>.stage.window.restore();`
 - ➔ `<nativewindow>.stage.window.close();`



Working with Native Windows

- ✓ Add custom sizing to a chromeless window:
 - ➔ best practice to use a custom component
 - ▶ `<nativewindow>.stage.window.startResize()`
 - ▶ `<nativewindow>.stage.window.startMove()`
- ✓ Hands on: create custom sizing and closing on a native windows.



There is no file in the AIR

following a native path



Working with the file system

- ✓ using the File class
 - ➔ A File object represents a path to a file or directory
- ✓ you can access the documents directory directly:
 - ➔ `File.documentsDirectory.resolve('path');`
- ✓ you can access the user directory directly:
 - ➔ `File.userDirectory.resolve('path');`
- ✓ Also: `desktopDirectory`, `applicationResourceDirectory` (where installed), ...



Working with the file system

- ➔ FileStream operations
 - ▶ Synchronous version
 - copyTo, deleteDirectory, deleteFile, listDirectory, moveTo, moveToTrash
 - ▶ Asynchronous version
 - copyToAsync, deleteDirectoryAsync, File.deleteFileAsync, listDirectoryAsync, moveToAsync, moveToTrashAsync
- ➔ When using Asynchronous versions, eventListeners must be added to listen for complete event.



Working with the file system

- ✓ nativePath: platform specific path of a File object.
- ✓ Browse for a file:
 - ➔ File.openForRead()
 - ➔ File.openForWrite()
 - ➔ File.openForAppend()



Working with the file system

```
var fileToOpen:File = File.documentsDirectory;  
selectTextFile(fileToOpen);
```

```
function selectTextFile(root:File):void  
{  
    var txtFilter:FileFilter = new FileFilter("Text",  
    "*.as;*.css;*.html;*.txt;*.xml");  
    root.browseForOpen("Open", [txtFilter]);  
    root.addEventListener(Event.SELECT, fileSelected);  
}
```

```
function fileSelected(event:Event):void  
{  
    trace(fileToOpen.nativePath);  
}
```



Working with the file system

✓ Relative path between two files:

```
var file1:File = File.documentsDirectory.resolve("AIR Test");
```

```
var file2:File = File.documentsDirectory;
```

```
file2 = file2.resolve("Air/koen/mmug.txt");
```

```
trace(file1.relativize(file2));
```

```
RESULT: koen/mmug.txt
```



Working with the file system

- ✓ Canonical versions of file names:
 - ➔ Assume there is a file **MMUG.txt** on the desktop.
 - ➔ If we resolve to mmUG.txt ==> it works...
 - ▶ `var file:File = File.desktopDirectory.resolve("mmUG.txt");`
 - ➔ if we want to know the REAL name with correct capitalisation:
 - ▶ `file.canonicalize();`
 - ▶ Also useful to know the real name of "8.3" names on Windows.(eg: "C:\\KOEN~1";)



Example

✓ Example on directory manipulation

➔ Topics discussed:

- ▶ `createDirectory()`
- ▶ `CreateTempDirectory()`
- ▶ `CreateTempFile()`



Yeah we are landed !
Is the flight over now ?



Where to fly from here

✓ Beta 1 Adobe AIR Documentation

- ▶ Developing Adobe AIR Applications with Flex
- ▶ Adobe AIR Quick Starts for Flex
- ▶ Flex 3 Language Reference (which includes the combined Flex/Flash/AIR API)

➔ **labs.adobe.com/go/air**

➔ Friends of ED upcoming titles:

- Foundation AIR: Creating Desktop Applications with the Adobe Integrated Runtime (january 2008)
- AdvancED AIR with Flex / Flash (january 2008)



Thanks !

Presentation and exercises
<http://www.newmovieclip.com>

Flight instructor : Koen De Weggheleire - www.newmovieclip.com