



Adobe

Developer Tips for Building Flash and AIR Apps for Android Devices

Renaun Erickson | Developer Evangelist

CodeMash 2011

Me and You

Background

- Adobe (3+ yrs) engineer turned developer evangelist
- Developed web & mobile applications
- Apps across Android, iOS, and PlayBook

Questions

- Who has built a Flash or AIR application?
- Who has built an AIR for Android application?
- Who has built any kind of Android application?

Overview

- The Adobe Technology Stack
- Demos
- Adobe Runtimes

Questions

- Who has built a Flash or AIR application?
- Who has built an AIR for Android application?
- Who has built any kind of Android application?

The Adobe Technology Stack

Unmatched Solutions for Great Experiences Across Screens

Multiscreen
Content and Applications



FLASH PLATFORM

TOOLS



Flash Professional Flash Catalyst

Flash Builder

FRAMEWORK



Flex

SERVERS



Flash Media Server Family



LiveCycle

SERVICES



Flash Platform Services

CLIENTS



Flash Player



AIR



Creative Suite



ColdFusion

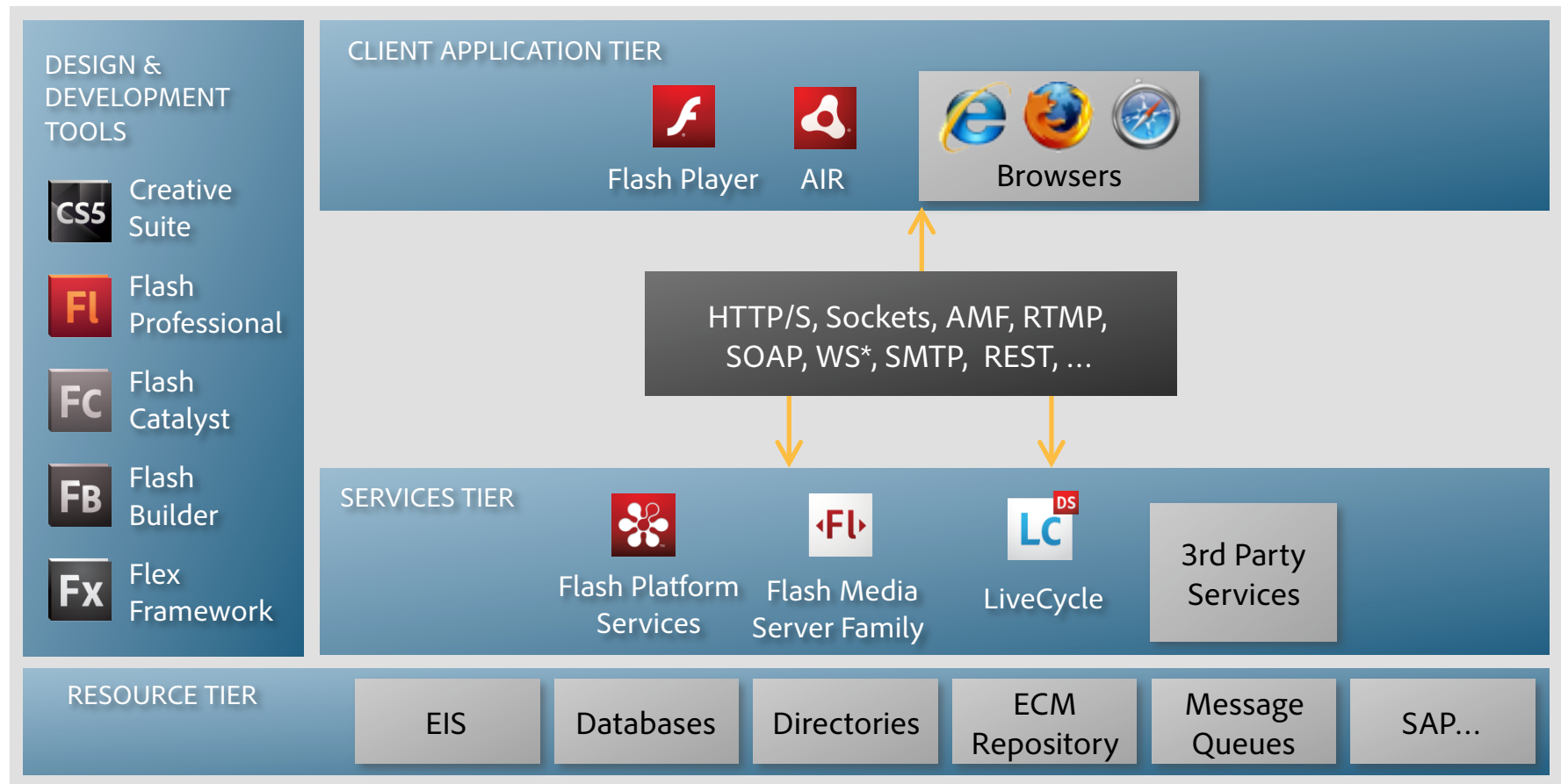


Analytics & Optimization

<http://www.adobe.com/flashplatform>

The Flash Platform integrates with existing technology stacks

The Flash Platform leverages data from any back-end system and exposes it in a rich, easy to use user experience.



Let's Play

Application Types: Web & Standalone

Flash Player 10.1 & AIR 2.5

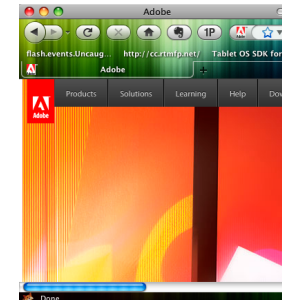
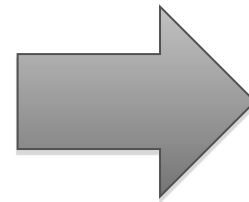
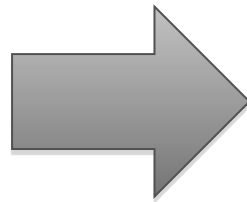


Flash Player Application

Compile

Deploy

```
SWF (width="3024", height="500", backgroundColor="#cccccc", frameRate="30");  
public class WebCastHelloWorld extends Sprite  
{  
    public function WebCastHelloWorld()  
    {  
        var helloButton:LabelButton = new LabelButton();  
        helloButton.label = "Hello World!";  
        helloButton.x = (stage.stageWidth - helloButton.width)/2;  
        helloButton.y = (stage.stageHeight - helloButton.height)/2;  
  
        myFormat:TextFormat = new TextFormat();  
        myFormat.color = 0x000000;  
        myFormat.size = 24;  
        myFormat.italic = true;  
        myFormat.align = "center";  
  
        var text:TextField = new TextField();  
        text.text = "Close";  
        text.setTextFormat(myFormat);  
  
        var closeButton:Button = new Button();  
        closeButton.addChild(text);  
        closeButton.addEventListener(MouseEvent.CLICK, closeWindow);  
        closeButton.x = (stage.stageWidth - closeButton.width)/2;  
        closeButton.y = helloButton.y - helloButton.height;  
        addChild(helloButton);  
        addChild(closeButton);  
  
        stage.nativeWindow.visible = true;  
    }  
  
    private function closeWindow(event:MouseEvent):void  
    {  
        stage.nativeWindow.close();  
    }  
}
```



ActionScript

SWF

HTML File/Web Site

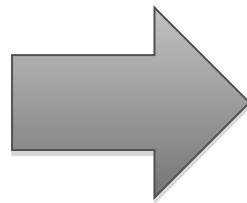
AIR Application

Compile

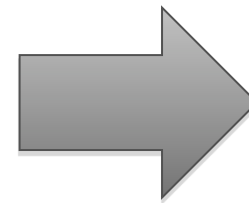
Package

```
SWF(width=3024, height=500, backgroundColor=#cccc, frameRate=30);  
public class WebCastHelloWorld extends Sprite  
{  
    public function WebCastHelloWorld()  
    {  
        var helloButton:LabelButton = new LabelButton();  
        helloButton.label = "Hello World!";  
        helloButton.x = (stage.stageWidth - helloButton.width)/2;  
        helloButton.y = (stage.stageHeight - helloButton.height)/2;  
  
        var myFormat:TextFormat = new TextFormat();  
        myFormat.color = #000000;  
        myFormat.size = 24;  
        myFormat.italic = true;  
        myFormat.align = "center";  
  
        var text:TextField = new TextField();  
        text.text = "Close";  
        text.setTextFormat(myFormat);  
  
        var closeButton:Button = new Button();  
        closeButton.addChild(text);  
        closeButton.addEventListener(MouseEvent.CLICK, closeWindow);  
        closeButton.x = (stage.stageWidth - closeButton.width)/2;  
        closeButton.y = helloButton.y - helloButton.height;  
        addChild(helloButton);  
        addChild(closeButton);  
  
        stage.nativeWindow.visible = true;  
    }  
  
    private function closeWindow(event:MouseEvent):void  
    {  
        stage.nativeWindow.close();  
    }  
}
```

ActionScript



SWF



AIR Application

AIR Runtime & mobileDevice Profile

AIR Runtime for Mobile Devices

- Current AIR Phrases – Safe using “AIR for X”
 - AIR for Android
 - Packager for iPhone
 - BlackBerry Tablet OS SDK for Adobe AIR
- AIR on Devices is not necessarily OS specific
 - Runtime is ported to a lot of OSes and chipsets/hardware
 - This does cause slight differences when hardware is different
- AIR APIs are limited by Profiles, then Hardware/OS

"desktop" Profile



NativeWindow
PDF, Printing
Context Menu
Notification
AS2 VM
DRM
WebKit

Capabilities

"mobileDevice" Profile



renderMode=GPU
Accelerometer
cacheAsBitmapMatrix
Orientation
SystemIdleMode
Camera Roll

SQLite
Fullscreen Mode
File System
Networking*
TLF Rendering
Camera/Microphone*

cacheBitmapMatrix
DRM
AS2 VM

"tv" Profile

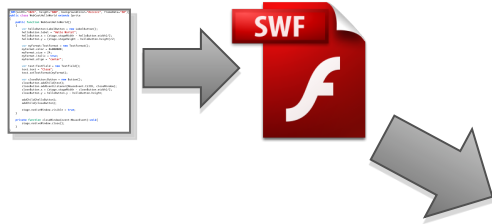
<http://bit.ly/aOIHQ3>

* May be present

The Process of Packaging AIR Applications

AIR Current Packagers

Compiling



Packagers



adt - Desktop/Android/TV uses AIR SDK
Creates: .air, .exe, .dmg, .apk



pfi - iDevice uses Packager for iPhone
Creates: .ipa



blackberry-airpackager - PlayBook
Creates: .bar

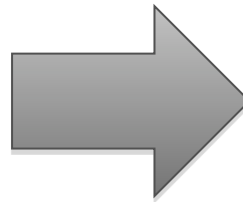
Same Application for AIR for Android

Compile

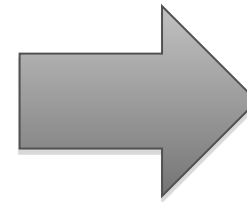
Package

```
SWF(width=1024, height=500, backgroundColor=#f0cccc, frameRate=30);  
public class WebCastHelloWorld extends Sprite  
{  
    public function WebCastHelloWorld()  
    {  
        var helloButton:LabelButton = new LabelButton();  
        helloButton.label = "Hello World!";  
        helloButton.x = (stage.stageWidth - helloButton.width)/2;  
        helloButton.y = (stage.stageHeight - helloButton.height)/2;  
  
        var myFormat:TextFormat = new TextFormat();  
        myFormat.color = 0x000000;  
        myFormat.size = 24;  
        myFormat.bold = true;  
        myFormat.align = "center";  
  
        var text:TextField = new TextField();  
        text.text = "Close";  
        text.setTextFormat(myFormat);  
  
        var closeButton:Button = new Button();  
        closeButton.addChild(text);  
        closeButton.addEventListener(MouseEvent.CLICK, closeWindow);  
        closeButton.x = (stage.stageWidth - closeButton.width)/2;  
        closeButton.y = helloButton.y - helloButton.height;  
        addChild(helloButton);  
        addChild(closeButton);  
        stage.nativeWindow.visible = true;  
    }  
    private function closeWindow(event:MouseEvent):void(  
        stage.nativeWindow.close();  
    )  
}
```

ActionScript



SWF



Android Application

adt - Android uses AIR SDK
Creates: .apk

Same Application for BlackBerry PlayBook

Compile

Package

```
[SWF{width="320", height="480", backgroundColor="000000", frameRate="30"}]
public class WebCastHelloWorld extends Sprite
{
    var helloButton:LabelButton = new LabelButton();
    helloButton.label = "Hello World!";
    helloButton.x = (stage.stageWidth - helloButton.width)/2;
    helloButton.y = (stage.stageHeight - helloButton.height)/2;

    var myFormat:TextFormat = new TextFormat();
    myFormat.color = "#000000";
    myFormat.size = 24;
    myFormat.italic = true;
    myFormat.align = "center";

    var textField:TextField = new TextField();
    textField.text = "Close";
    textField.setTextFormat(myFormat);

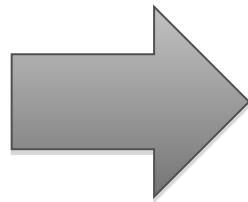
    var closeButton:Button = new Button();
    closeButton.addChild(text);
    closeButton.addEventListener(MouseEvent.CLICK, closeWindow);
    closeButton.x = (stage.stageWidth - closeButton.width)/2;
    closeButton.y = helloButton.y - helloButton.height;

    addChild(helloButton);
    addChild(closeButton);

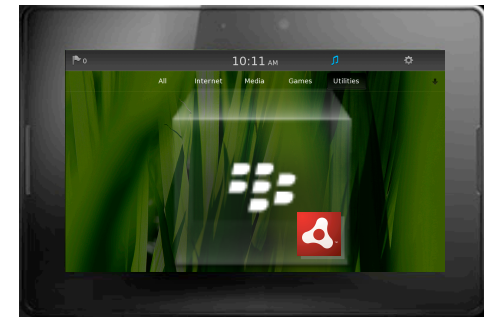
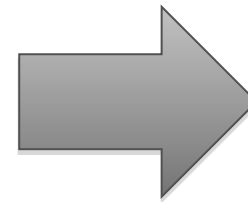
    stage.nativeWindow.visible = true;
}

private function closeWindow(event:MouseEvent):void{
stage.nativeWindow.close();
}
}
```

ActionScript



SWF



BlackBerry PlayBook
Application (.bar)

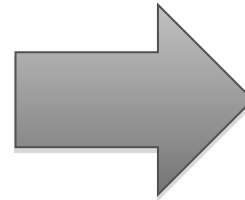
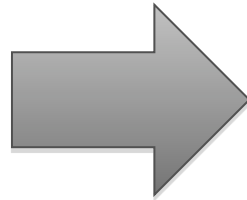
blackberry-airpackager – PlayBook
Creates: .bar

Same Application for iOS

Compile

Package

```
SWF(width=1824, height=1000, backgroundColor=#cccccc, frameRate=30);  
public class WebCastHelloWorld extends Sprite  
{  
    var helloButton:LabelButton = new LabelButton();  
    helloButton.label = "Hello World!";  
    helloButton.x = (stage.stageWidth - helloButton.width)/2;  
    helloButton.y = (stage.stageHeight - helloButton.height)/2;  
  
    var myFormat:TextFormat = new TextFormat();  
    myFormat.color = #000000;  
    myFormat.size = 24;  
    myFormat.italic = true;  
    myFormat.align = "center";  
  
    var text:TextField = new TextField();  
    text.text = "Click";  
    text.setTextFormat(myFormat);  
  
    var closeButton:Button = new Button();  
    closeButton.addChild(text);  
    closeButton.addEventListener(MouseEvent.CLICK, closeWindow);  
    closeButton.x = (stage.stageWidth - closeButton.width)/2;  
    closeButton.y = helloButton.y - helloButton.height;  
    addChild(helloButton);  
    addChild(closeButton);  
    stage.nativeWindow.visible = true;  
}  
private function closeWindow(event:MouseEvent):void{  
    stage.nativeWindow.close();  
}
```



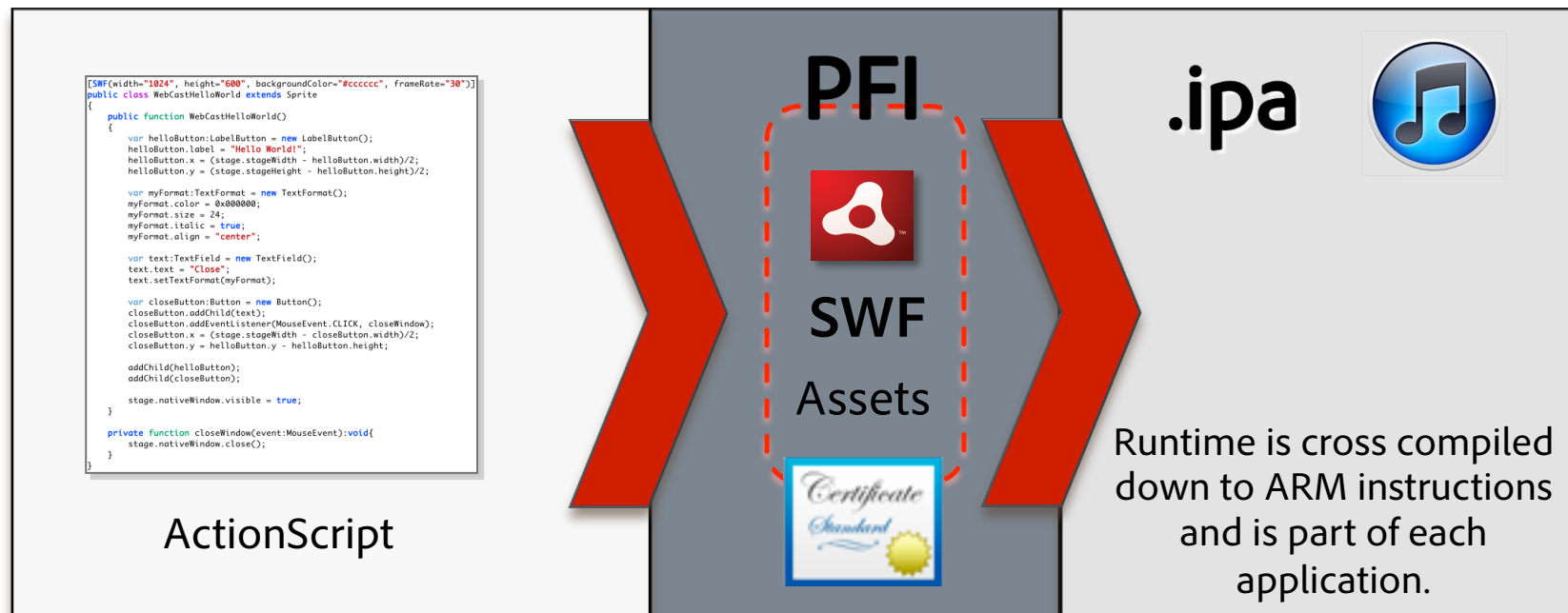
ActionScript

SWF

iOS Application (.ipa)

pfi- iDevices
Creates: .ipa

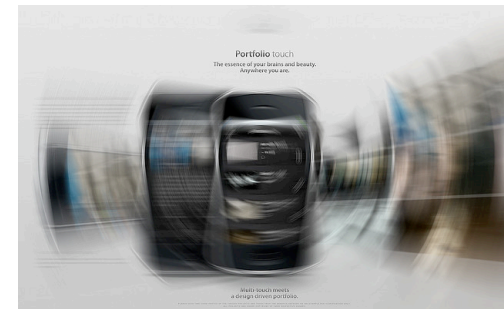
How Packager for iPhone (iOS) Works



Multiscreen Landscape

An Adobe Developer's View

- Adobe Runtimes on Form Factors
 - Desktop
 - Mobile
 - Tablet
 - Digital Home (TV, Set-top boxes)
 - Consoles
- Hardware Constraints
 - CPU Varies (600Mhz to 1Ghz +)
 - ARM (Android, iOS, etc..), x86 (Mac, Win, *nix)
 - GPU Varies (OpenGL ES 1.1 to 2.0 +)
 - Memory Varies (128MB to 512MB +)
 - Screen Quality and Sizes
- User Expects Great Experiences
 - High Quality Assets
 - Fast Input Responses

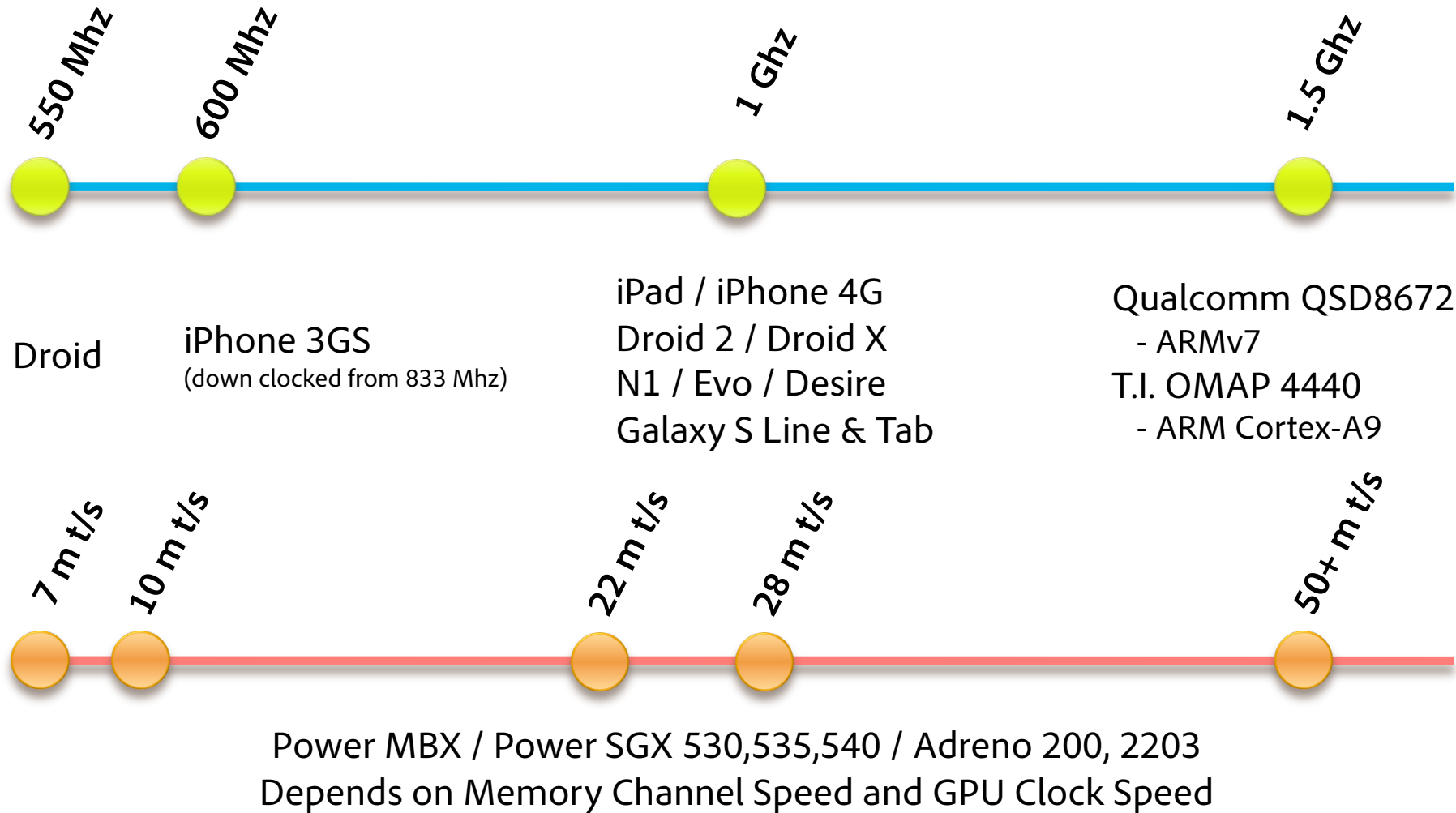


Physical Size / Pixels Per Inch

Screen Resolutions are Independent of Physical Screen Size

Device	Resolution	PPI	Physical
Nexus One / Droid Incredible / HTC Desire	800x480	254	3.7'
HTC Evo 4G / HTC Desire HD	800x480	217	4.3'
Droid / Droid 2	854x480	265	3.7'
Droid X	854x480	228	4.3'
Samsung Galaxy S Vibrant	800x480	232	4.0'
iPhone 3GS and lower	480x320	163	3.5'
iPhone 4	960x640	326	3.5'
iPad	1024x768	132	9.7'
Galaxy Tab/PlayBook	1024x600	170	7'

CPU / GPU



*m t/s = millions of triangles per second
<http://alienbabeltech.com/main/?p=19309>

Other Hardware Considerations

- Touch Screen
 - Different types perform differently
 - Max number of touch points vary
- Memory RAM / ROM
 - RAM Varies from 128MB to 768MB
 - Default ROM small to 1GB+
 - Expanded by SD or internal storage
 - Type of Memory affects GPU speeds

Expect performance like PCs of 7 years ago!

Developer Tips

Using Capabilities to Develop for Different Devices

Capabilities.screenDPI

Capabilities.screenResolutionX

Capabilities.screenResolutionY

Multitouch.supportedGestures

Multitouch.supportsGestureEvents

Multitouch.supportsTouchEvent

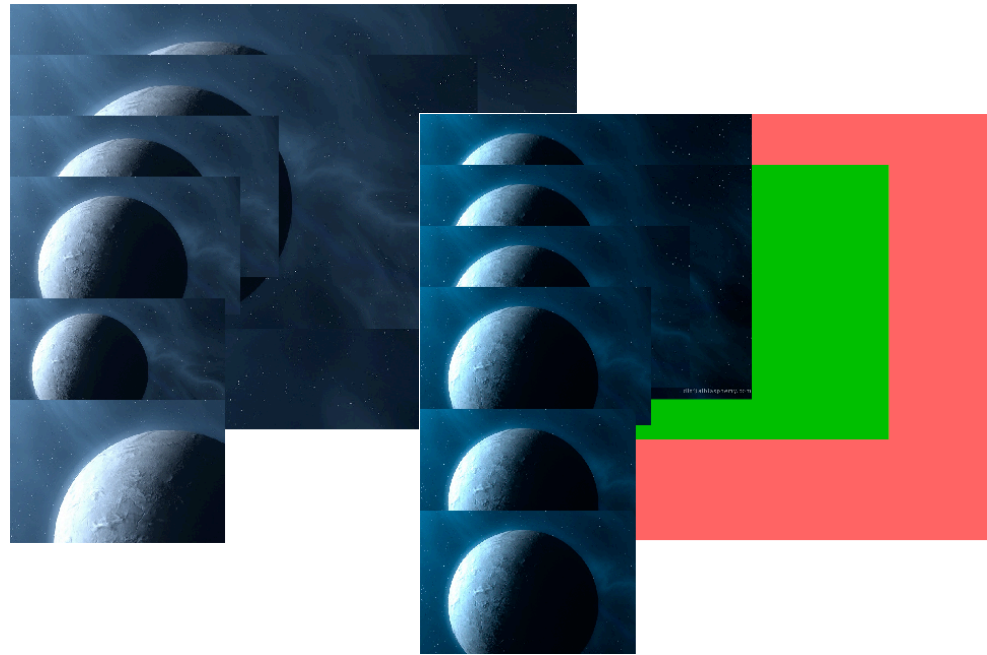
Geolocation.isSupported

Accelerometer.isSupported

Examples



Accelerometer for Pitch
if Device Supports it



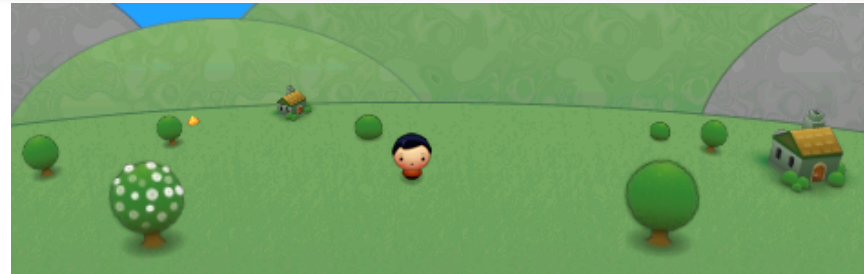
Scale Content Based on
DPI or DPI/Physical Size
Combo

Performance GPU and Memory

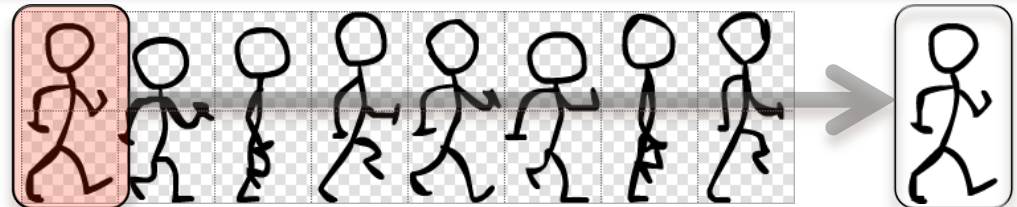
GPU Bitmap Transformations

- (cacheAsBitmapMatrix)

Object Pooling



Bitmap Blitting



Multiuser Applications

Using Flash Platform Services to Build Collaborative Applications

LiveCycle Collaboration Services

P2P – RTMFP Support

Stratus – beta bootstrap service

FMS 4.0 Enterprise Server / Dev license

Multicast with NetConnection, NetGroup, and GroupSpecifier

```
var groupSpec:GroupSpecifier = new GroupSpecifier("myGroup/groupOne");
groupSpec.postingEnabled = true;
groupSpec.ipMulticastMemberUpdatesEnabled = true;
groupSpec.addIPMulticastAddress("225.225.0.1:30303");

group = new NetGroup(nc,groupSpec.groupSpecWithAuthorizations());
group.addEventListener(NetStatusEvent.NET_STATUS,netStatus);
```

Using Multitouch

- Set input method to which type of touch method
 - `MultitouchInputMode.GESTURES` – Gestures
 - `Multitouch.supportedGestures`
 - `Multitouch.supportedGestureEvents`
 - `MultitouchInputMode.TOUCH_POINT` - Raw Touch Point
 - `Multitouch.supportedTouchEvent`
 - Can build your own gesture library with raw data points?
 - <http://gestureworks.com/> - Commercial Product

Having Fun with Microphone

```
private function start():void
{
    mic = Microphone.getMicrophone();
    mic.rate = 44;
    mic.setSilenceLevel(5, 100);
    mic.addEventListener(SampleDataEvent.SAMPLE_DATA, dataHandler);
}
```

```
private function dataHandler(event:SampleDataEvent):void
{
    findFrequency(event.data);
}
```


Accelerometer

```
if (Accelerometer.isSupported)
{
    acc = new Accelerometer();
    acc.addEventListener(AccelerometerEvent.UPDATE, accUpdateHandler);
}
```

```
private function accUpdateHandler(event:AccelerometerEvent):void
{
    pitch = int(event.accelerationY*90);
    if (pitch < 0)
        pitch = 5;
    pitch = 90 - pitch;
}
```

- **Adobe Developer Connection for Mobile Resources**
- - <http://www.adobe.com/devnet/devices/fpmobile/>
- **AIR for Android**
- - <http://www.adobe.com/products/air/>

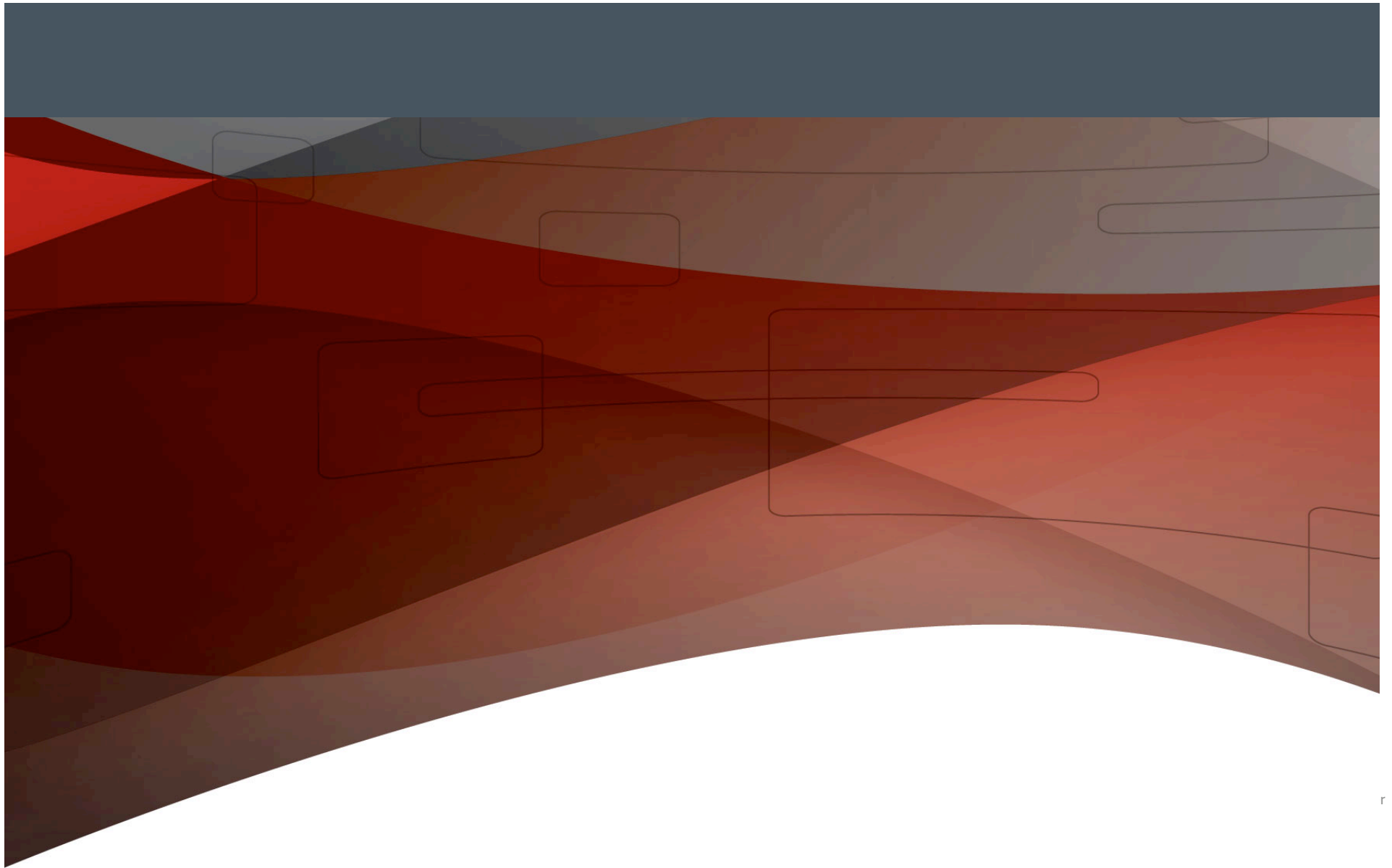
Twitter: @renaun

Blog: <http://renaun.com>

Email: renaun@adobe.com



Adobe



Adobe, the Adobe logo, Adobe AIR, the Adobe AIR logo, ActionScript, AIR, Flash, Flash Builder, Flash Catalyst, the Flash logo, Flex, and LiveCycle are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners. Printed in the USA. 91023957 3/10