

## ***Integration of Adobe Video Editor in Android Programming: Using Flash CS5 for Synchronized Android Development***

***Sonika Das, Indrani Mukherjee***

***Bengal College of Engineering & Technology, Durgapur, West Bengal***

***E-mail: indranimukherjee1982@gmail.com***

### ***Abstract***

*This paper details how Adobe video editing can be integrated gainfully with Android Development. As no other industry is advancing at the pace at which mobile industry is, developers need to keep upgrading their skills constantly. Mobile connectivity is no longer just about texting, sending MMS and emailing.*

*It includes games, map applications, video conferencing, online video streaming, and more. This paper explores the methods of integration of the Adobo Video and Animation editor, Flash CS5, with Android Programming. This is essential for Android developers of today, simply because of the rapid strides that mobile technology development is taking.*

***Keywords:*** *Android Programming, iOS, Flash CS5, Android Development*

### **INTRODUCTION**

Today, there are 5 billion people around the world using mobile phones. It is a staggering figure. As a frame of reference, there are only 1.7 billion PCs being used around the world. Three years ago, a smart phone allowed you to send e-mail. Today, when you think smart phone, you think e-mail, web, games, MMS, video conferencing—you think of a computer in

your pocket. There are a number of companies leading the next wave of smart phone market. Google, Apple, RIM, Nokia, Microsoft, and HP (with Palm) all have their own operating systems and hardware. It seems almost every three to six months these companies leapfrog each other. Consider this—at the end of 2009, a mobile phone running at 500 MHz with a 3 MB camera was considered screaming

fast. Now, you can pick up those same phones for less than \$100. If you want something faster you go for 1 GHz with a 1 GB of RAM, an 8 MP camera, front and rear facing cameras, proximity devices up the wazoo, and sophisticated operating systems (OS) that rival, and in some cases exceed, what you can accomplish on your desktop.

This is not a mobile phone in your pocket. It is a screamingly fast computer. With this all said, the smart phone market is still very small. You can take all the iPhones, Android phones (Figure 1.2), and BlackBerrys and you will have less than 300 million devices worldwide. With a global figure of 5 billion mobile users, it is clear that the smart phone market has massive potential for growth. So, what does it mean to develop for a smart phone? At the end of the day, there are essentially two ways you can develop for a smart phone:

- Develop directly to the software development kit (SDK)
- Develop using an intermediate technology

Each mobile device comes with an SDK that you can use for development. An SDK comes with the development tools, –bundling tools, and emulators you need to test your code. When you need access

to the latest and greatest technology, you need to use an SDK.

The challenge you have with using core SDKs is that you need to use the native development language. This is different for each SDK. For instance, Apple prefers you use Objective-C whereas Google prefers you use Java. The second way to develop mobile devices is to use an intermediate technology that allows you to build for multiple devices using only one language. An example of this is the 3D game development technology called Unity 3D. Unity uses JavaScript to let you to script your games and then converts the JavaScript into code that will allow you to build iPhone, Android, and Windows Desktop applications. The downside to using intermediate technologies is that you are dependent on the development company to update their tools to the latest SDKs and technologies. This can be hard work as the SDKs are frequently updated. For instance, Apple has updated its iOS operating system four times in three years, and Google’s Android has been updated five times in less than two years. With that said, it is much easier to develop using intermediate languages. You can leverage skills you already have without having to go through the learning curve of adopting a new language. In May 2010 at the Google I/O conference, Adobe announced

that it would be bringing both the Flash Player and AIR (Adobe Integrated Runtime) to Google's Android 2.2. This was really big news for Flash developers for several reasons:

1. The version of Flash coming to the Android is the latest 10.1 version, not some crippled alternative.
2. AIR gives you an immediate in-road into mobile device development, leveraging the tools and knowledge you already have.
3. Flash is coming to 19 other mobile device companies.
4. Android runs on tablets and TVs as well as phones

The Flash Player that is now available for all Android 2.2 users, shown in Figure 1.3, is very efficient. There has been a lot of noise from companies such as Apple stating that Flash is a battery hog and will kill your phone's CPU. Is this true? The reality is that it is not. Tests have been conducted showing that the Flash Player on mobile devices is highly efficient and does not cause the CPUcripling results Apple is stating. The Flash Player works inside the browser in Android. You trigger the use of the Flash Player by tapping on

the Flash content in the web page. For instance, you can view a Hulu.com video by tapping on the content in the page. Adobe's modification of AIR for mobile devices was the really big story at Google's I/O. AIR is a very powerful, mobile technology. It reaches for the same goals that Java set in the 1990s: write once, run anywhere. Unlike Java, AIR really achieves its goal. AIR apps are built in the Flash Professional development environment using Flash technologies you are already used to, such as ActionScript to program your solutions, MPEG video for video, and the same animation techniques you have been using for years. Adobe's support for Flash on mobile devices will be coming to 19 other mobile development companies. This means that the techniques you learn in this paper will be applicable beyond just Android. Other entities that will be adopting Adobe's technologies include RIM's BlackBerry, Nokia, HP/Palm WebOS, and Microsoft's Windows Phones Series 7. Notably absent is Apple, Inc. The year 2010 will go down as the year that Apple drew a line in the sand and said very publicly, "We will not support Adobe's Flash." It is a shame that Apple has made this stance, since Apple's iOS is a very important part of the mobile market. Let's hope it changes its mind. Earlier I mentioned how rapidly the

mobile market is growing. Today, that market is predominantly composed of phones; but there are additional tools joining this market. During 2010, Apple released the massively popular iPad, a tablet computer that is very mobile and very light. Not to be outdone, rival companies, such as Nvidia, Samsung, Dell, and Cisco, are also coming out with their own tablets. The devices range in size from 5 inches all the way up to 11 inches and beyond. What they all share is that they are running Android as their OS. They come pre packaged with support for Flash. Another device that is coming out of the mobile world is Google TV. At its essence, Google TV is really a modified version of Android that runs directly on your TV. And, yes, your Flash apps will run here, too. Last year you could run your Flash applications only on Windows and Mac computers. Today you can add smart phones, tablets, and TVs. Your Flash can literally go with you wherever you want to go. It is not all roses, of course. To get your Flash apps running in AIR on all these new devices, Adobe did choose to make one big change: You must develop your solutions using ActionScript 3.0 (AS3). AS3 has been around since 2006. If you have not made the jump to AS3, then I will help you as we step through this paper . You can no longer leverage the

older AS1 and AS2 scripts that you have been using for years. Time to start fresh. The first section of the paper explains how a Flash designer can set up a Flash CS5 environment to publish Android apps. Later you will step through the process of downloading, installing, and running the Android SDK, necessary for your development. By the end of this section you will have created your first Android application using Flash CS5. At the end of this paper you will have the knowledge to build almost any type of Flash-based application for the Android OS 2.2 and greater. How cool is that? So, let's get started.

## **DESIGNING AND DEVELOPING FOR ANDROID HARDWARE**

Before we get involved with setting up your design and development environment, let's take a little time to review how you should approach developing applications that run on an Android device. There are a number of design considerations you always want to keep in the back of your mind as you work on your apps. They are:

- Different hardware
- Hardware acceleration
- Touch interaction

An Android phone is simply very different than a desktop, and you need to develop your app to take advantage of these differences.

Working with Android Hardware the Android platform has been available for less than two years. In that time it has gone from being available on a few phones to being installed on dozens of different phones available on almost every mobile carrier. Today there are over 50 different mobile phones running Android OS 1.5 and greater. Flash is supported on all phones that run Android 2.2. Table 1 gives you a list of the Android phones that currently support Flash and AIR. The table is broken down by manufacturer, name of the phone, screen display (where available), and additional notes about the phone.

You can see from the devices listed in Table 1 that there is a broad range of hardware specifications for Android phones.

The number one hardware difference you will need to keep in mind is screen size. The default screen size for Android development is  $320 \times 480$  ppi (points per inch) but, as you can see from the list, this is not always the case. Screen resolutions

range from  $240 \times 320$  for the HTC Wildfire all the way up to  $854 \times 480$  for the Motorola Droid X. How do you design apps for this broad range? The trick comes in how you use Flash to do the work for you. We will be getting into that in more detail as you work your way through the paper. Just keep in mind that not all Android phones are created equal.

In addition to screen size, the second feature that you will find different from one device to another is RAM and CPU. The more RAM you have determines how much data can be crunched with active apps.

The multitasking feature in Android allows for six core apps to be running simultaneously. But you may have many more utilities running. To run more applications will require more RAM. Current smart phones have 256 MB of RAM, with others having as much as 1 GB of RAM. Future devices will have RAM levels that rival desktop computers. For now, however, develop applications that carefully manage the amount of RAM you use.

The CPU listed earlier is slightly misleading. Almost all smart phones are developed with a system-on-chip design (SOC). An SOC merges the CPU, GPU,

RAM, and other systems into one chip. This architecture is typically built on ARM CPUs. The ARM architecture is highly energy efficient, allowing mobile phone batteries to last longer. Intel,

Nvidia, and AMD are also starting to join the ultra-efficient mobile chip market.

**Table 1: Android Phones That Support Flash and AIR**

Manufacturer	Name	Display	Notes
Acer Inc	Liquid E	320 × 480	Smart phone with underclocked 768 MHz Snapdragon processor.
Acer Inc	Liquid E Ferrari	320 × 480	A customized version of Liquid E with Ferrari visual styling.
Acer Inc	beTouch E400	320 × 480	SIM-free smart phone with a 600 MHz CPU, 3.2" resistive touch screen, and a 3.1 MP camera.
Acer Inc	Liquid Stream (S110)	320 × 480	1 GHz SnapDragon CPU, 3.7" AMOLED WVGA capacitive touch screen, 5 MP camera.
Dell	Thunder	320 × 480	4.1" WVGA OLED screen, and an 8 MP camera.
Dell	Flash	320 × 480	3.5" WVGA LCD screen, 5 MP autofocus cam, 512 MB of RAM and ROM with microSD expansion up to 64 GB, WiFi, TV-out, an 800 MHz Qualcomm MSM7230 processor.
Dell	Smoke	320 × 480	2.8" QVGA touch screen, 5 MP autofocus cam, microSD expansion to 32 GB, WiFi, Bluetooth, and an 800 MHz Qualcomm MSM7230 processor.
HTC Corporation	HTC Aria	480 × 320 (HVGA) 3.2"	A mid-range AT&T exclusive, running on Android 2.1 with HTC Sense; uses 600 MHz MSM 7227 processor, 5 MP camera; similar to HTC Legend.
HTC Corporation	HTC Desire	480 × 800 (WVGA) 3.7"	Similar to Nexus One but adds HTC's Sense UI Optical trackpad and Hard buttons, but does not have dual microphones as the Nexus One.
HTC Corporation	HTC Hero, HTC Droid Eris, T-Mobile G2	320 × 480 3.2" 180 ppi	The Hero has two design versions. The original design is similar form factor to the Magic; the U.S. release design is more curved at the edges and has the controversial "chin" removed. Both use HTC's customized UI called HTC Sense, which looks considerably different compared to HTC Dream and Magic phones.
HTC Corporation	Droid Incredible	800 × 480 3.7" AMOLED	Successor to the HTC Droid Eris; sports an 8.0 MP camera with dual-flash LED, FM radio tuner, and 8 GB onboard flash memory, 3.7" AMOLED screen, native resolution of 480 × 800 px.

*Continued*

**Table 2: Android Phones That Support Flash and AIR-continued**

Manufacturer	Name	Display	Notes
HTC Corporation	HTC Legend	480 × 320 (HVGA) 3.2" AMOLED	Announced at Mobile World Congress 2010 in Barcelona.
HTC Corporation	HTC Evo 4G (formerly HTC Supersonic)	480 × 800 4.3" 217 ppi	A high-end Android phone, includes the HTC Sense UI, similar form factor to the Droid Incredible and HTC HD2. Contains many advanced phone features, including an 8 MP rear-facing camera along with a 1.3 MP front-facing camera. The Evo 4G is currently (as of 5/22/10) the only phone to offer 4G Internet access (currently using Clearwire WiMAX).
HTC Corporation	Google Nexus One, Codenamed HTC Dragon, HTC Passion	480 × 800 (WVGA) 3.7" 252 ppi	The first phone to be sold directly by Google, the Nexus One was initially available exclusively online, unlocked. It can now be bought on subsidized contract with various networks.
HTC Corporation	myTouch 3G Slide	320 × 480 (HVGA) 3.4"	5 MP camera, QWERTY four-row keyboard, and a Swype on-screen keyboard.
HTC Corporation	HTC Wildfire	240 × 320 (QVGA) 3.2"	5 MP autofocus camera with LED flash, 802.11b/g WiFi, GPS/AGPS, Bluetooth 2.1+EDR, 512 MB Flash and 384 MB of RAM, microSD expansion.
Motorola	Motorola Droid, Motorola Milestone worldwide GSM version	854 × 480 3.7" 265 ppi	
Motorola	Motorola Droid X	854 × 480 4.3"	
Motorola	MOTO XT720, Motoroi, Motorola Milestone XT720	320 × 480	8 MP camera(Flash), HDMI, FM radio, T-DMB, available only in Korea.
Pantech	Sirius Sky	480 × 800 (WVGA) 3.7"	1 GHz Snapdragon processor, 3.7" (WVGA 480 × 800, AMOLED), 5 MP camera, WiFi, Bluetooth, GPS, and microSD expansion.
Pantech	Sirius Izar	480 × 800 (WVGA) 3.7"	Qualcomm MSM7227(600 MHz), 3.2" (WVGA 480 × 800, LCD), 5 MP camera (AF), WiFi, Bluetooth T-DMB and microSD expansion.

Continued

**Table 3: Android Phones That Support Flash and AIR-continued**

Manufacturer	Name	Display	Notes
Pantech	Sirius Alpha	480 x 800 (WVGA) 3.7"	Minor upgrade of Sirius Sky.
Samsung Group	Galaxy A	3.7" AMOLED	5 MP camera, T-DMB, GPS, Bluetooth, 802.11n Wi-Fi, and video calling. Will be available only in South Korea.
Samsung Group	i9200	320 x 480	4.3" AMOLED 1280 x 720, 2 GHz CPU, 1 GB RAM, 4 GB ROM, 32 GB flash, microSD oraz 8 MP primary camera + 2.0 secondary camera
Samsung Group	Galaxy S	480 x 800 (WVGA) 4.0" Super AMOLED	1 GHz processor

At first, the original SOCs in the 2007/2008 smart phone market were very slow compared to a PC. Today, however, it is common to have a 1 GHz CPU/SOC with 2 GHz multicore SOCs shipping 2011. For a good Flash/AIR experience you need to be running a 1 GHz CPU/SOC architecture. The Motorola Droid runs at 500 MHz, and can run Flash, but you are better off testing with an HT Evo, Nexus One, or Motorola Droid X, all of which run at 1 GHz or faster.

**ANDROID HARDWARE ACCELERATION**

Phones come loaded with technology in the hardware. This is awesome for you as a developer. Following are some key hardware technologies that you will want to keep in mind as you develop for the Android OS:

- Touch-sensitive screen
- Sound/microphone
- Vibration
- Camera
- GPS
- Accelerometer/compass

The touch-sensitive screen seems like an obvious hardware feature, but it is your main input to your device and you use your finger. More on that in a moment.

Every Android phone supports audio, both to listen through speakers and to record with a microphone. We will cover audio in more detail later in the paper , but you will want to keep your audio files in MP3 and WAV format. Unlike desktop computers, where you cannot guarantee if there is a microphone installed by the manufacturer, you can guarantee that there

is a mic on every Android device. Why? It's a phone! You need one to speak through when you make calls.

Haptic feedback is the method by which you can provide vibrations to the end user through the phone. This is good for situations where the audio is turned off; the app is designed for the deaf or hard of hearing. You can also add vibration to games to add to the overall sensory experience.

All Android phones come with a camera, with many of them supporting video and LED flash.

A Global Positioning System (GPS) allows you to create solutions that are dependent on location. Want to develop an app that shows you the movie theater nearest to your current location? Use GPS to do the location work for you.

Accelerometers and compass hardware detect when the phone is being moved and in which direction along three distinct axes (X, Y, and Z). Expect newer phones to start shipping with gyroscopes to add three more axes (pitch, yaw, and roll). These three hardware features give you pinpoint control over your game

development. Think Wii Remote, but for phones.

These hardware features highlight the main tools you use from the Android hardware. Adobe's Flash will interact with all of these special hardware features.

## **TOUCHING YOUR APPLICATION**

In the previous section we talked about using touch as the way you interact with Android. This is a very important concept as you look to develop solutions for the Android OS. No matter what Steve Jobs will tell you, a finger is simply not as accurate as a stylus. It is, however, much more convenient to use.

In addition to the actual size of the input area you develop in your solutions, you will also want to bear in mind that not all touch screens are created the same. The apps you will create in this paper are designed to work on all touch screens, no matter what the device. If, however, your application requires very accurate touches and gestures, such as swipe, then you will want to test on several of the devices listed earlier. This is because vendors use different touch screens. Some are more accurate than others.

## **CONFIGURING THE ANDROID SDK PUBLISH SETTING**

The core of your development completed in this paper requires only the use of Flash CS5. To test your applications you will need to have the Android SDK installed on your computer. The SDK will allow you to complete the following:

- Run command line Flash build tools
- Test your Android App in an emulator
- Install your Android onto a physical device

Installing the Android SDK is not something to fear; you just need to follow the steps. For the most part, Google has made this a rather painless experience. This section will take you through what you need to do to install Android onto your development computer.

The first thing you need to do is see if your development environment will support the Android SDK. The minimum requirements are:

- Windows XP (32-bit) or Vista (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (tested on Linux Ubuntu Hardy Heron)

You will want to also ensure that your computer is fairly fast. I run my

development on an iMac with 4 GB of RAM and 2.2 core duo CPU. This works great for me. If your computer is new within the last three years, then you should be fine. In addition, you will need about 300 MB of hard drive space to install all the software.

The next step before you even get to installation is to ensure that you have the latest Java Developer Kit (JDK). Version 5 or 6 will work. You can download the latest JDK at <http://java.sun.com/javase/downloads/index.jsp>. The Android SDK will not install if you do not have the JDK installed. Now, to download and install the Android SDK:

1. Start by going to <http://developer.android.com/sdk/index.html> and downloading the latest SDK release (a ZIP file; Figure 1).
2. Save the SDK to your hard drive.
3. From the root of your computer, create a folder called Developer. Unzip the files from the Android SDK to this folder. There will be a lot of files.

4. Open the Developer folder. Click the subfolder called Android and then the folder called Tools.
5. Double-click the file called Android to access the Android SDK and AVD manager. The role of the Android SDK manager is to allow you to download and install Android SDK releases (Figure 2).
6. When the Android SDK and AVD Manager opens, select Available Packages on the left screen. You should see a link to an XML file with a check mark to the left of it. If you do not see anything, choose the Refresh button.

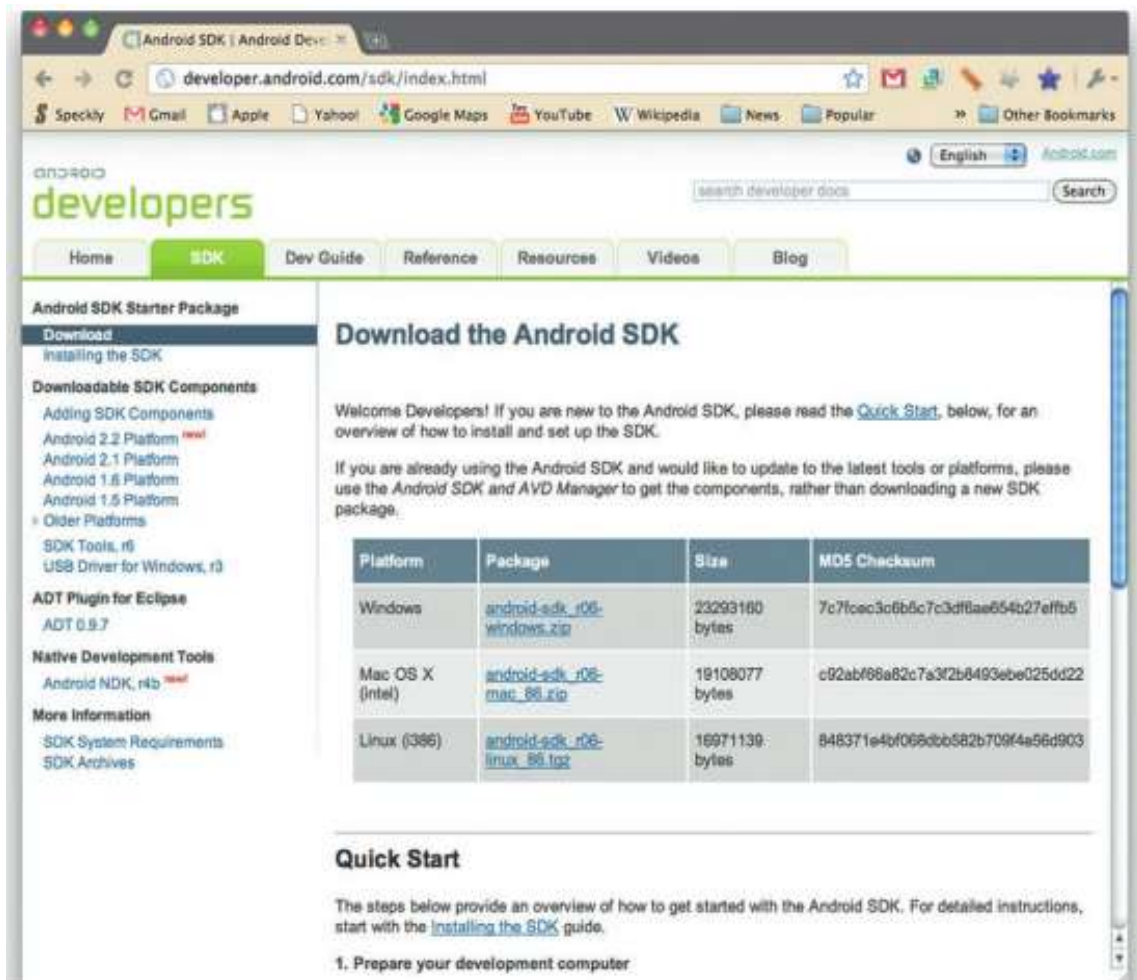
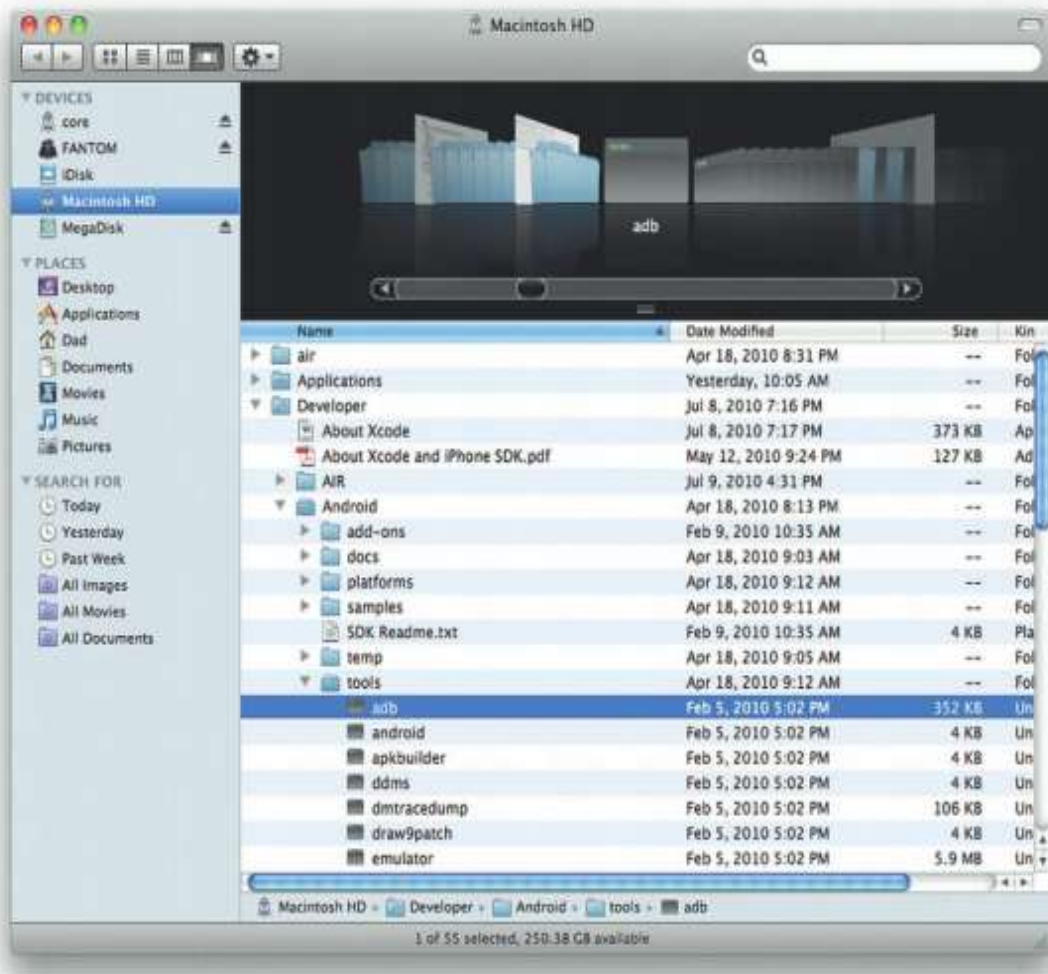


Figure 1: Google's Android SDK site



**Figure 2: Extract the Android SDK to your hard drive.**

7. Select the check mark
  8. The Android SDK will check which SDKs are available for testing. You will see SDKs for Android Platforms 1.5, 1.6, 2.1, and 2.2.
  9. Select the check marks alongside the Android 2.2 SDK. Choose the Install Selected button.
  10. The files will download and install onto your computer. This may take some time depending on the speed of your Internet connection.
- When you have completed the installation you will have all the files and tools needed to test your Flash applications in Android simulators or on a physical Android device.

## SETTING UP FLASH CS5 FOR ANDROID DEVELOPMENT

Adobe is going to great lengths to make it very easy for you to develop Android applications. To this end, you can now develop Android applications with either Flex or Flash CS5 Professional. Flex development requires the use of a command line utility to make the final Android application.

Everyone hates command lines. So, to make life easier, you can stick with Flash CS5 Professional. This is how we will be building applications for Android throughout this Paper.

Flash CS5 does not ship with native support for Android. Ironically, it does ship with support for iPhone development—but you cannot publish to the Apple iTunes Store. Trust me, I have tried and failed.



*Figure 3: Extract the Android SDK to your hard drive.*

You will need to install the tools needed for Flash CS5 to create files for Android. Follow these steps to install the Android tools you need:

1. You must be running the latest release version of Flash CS5 Professional. Run Adobe update tool to ensure that you have the latest patches and updates.
2. If you have Flash CS5 running you will want to close it.
3. Go to Adobe's AIR for Android web page at <http://labs.adobe.com/technologies/air2/android/> and download the MXI file.
4. Open the Adobe Extension Manager installed on your computer. Open the MXI file and choose install (Figure 3).

That's it. You now have all the tools needed to build your first Android application with Flash CS5 Professional.

## **INSTALLING YOUR AIR APPLICATION ONTO AN ANDROID OS**

Flash uses AIR to create your applications for Android. By default, AIR is not installed on the Android phone. This does

not stop you from installing your new app, it will simply stop you from running it.

Fortunately, AIR is freely available in the Google Marketplace. If your phone does not have AIR installed you will be prompted to download and install it from the Marketplace. There are no complex hoops to jump through. If you have installed one app, then you know how to install AIR and enjoy all the Flash apps in the Marketplace.

## **BUILDING YOUR FIRST APPLICATION FOR ANDROID USING FLASH CS5**

The goal for your first Android application is a simple one: to get a basic Flash movie running successfully on your Android phone. The following steps will take you through the whole process, at the end of which you will have your first native Android application created using Flash tools. The next section explains how to install the Android Application onto your device.

1. Begin by opening Flash CS5. Select File → New to open the new file window.
2. Select Template from the top button of the new file window (figure 4).



*Figure 4: Flash CS5 has a template for Android applications*

3. Choose AIR for Android from the left category window. On the right hand side you will see 480x800Android. Select OK (Figure 5).

4. To keep things simple, we are going to create all we need for a simple test. Save your file to your hard drive. Name the file FirstApp fla.

5. On the Stage use the text tool to draw a text region. Android does not support the

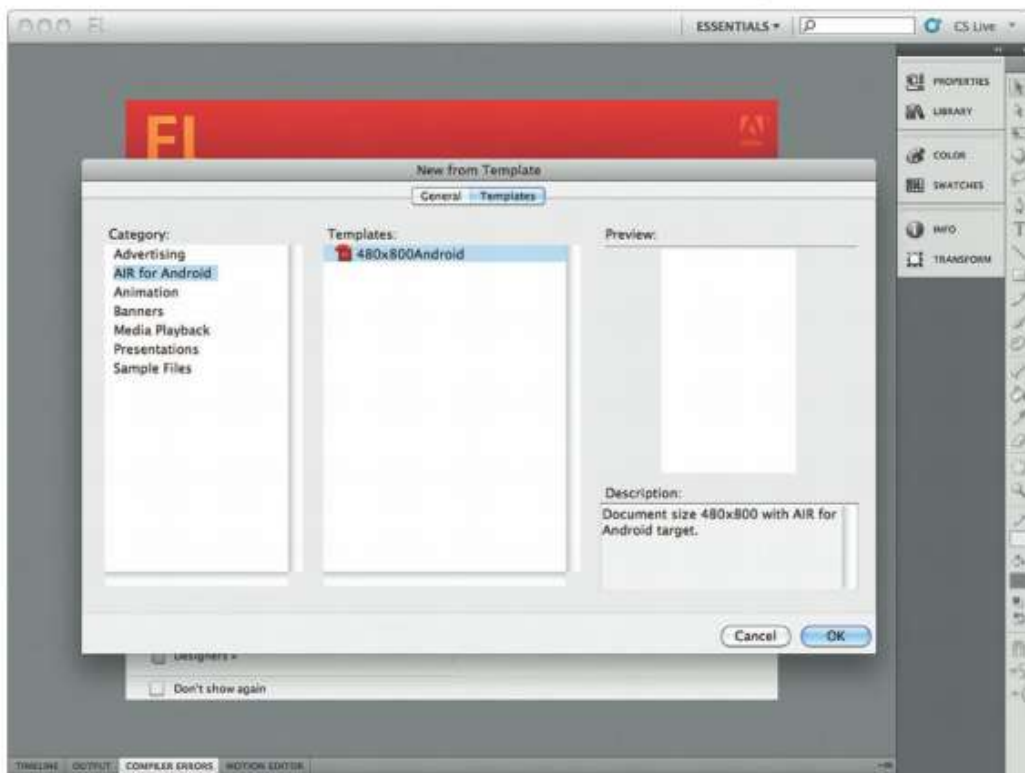
new TLF text. Change the text format to Classic Text (Figure 6).

6. Set the font Family to \_sans.

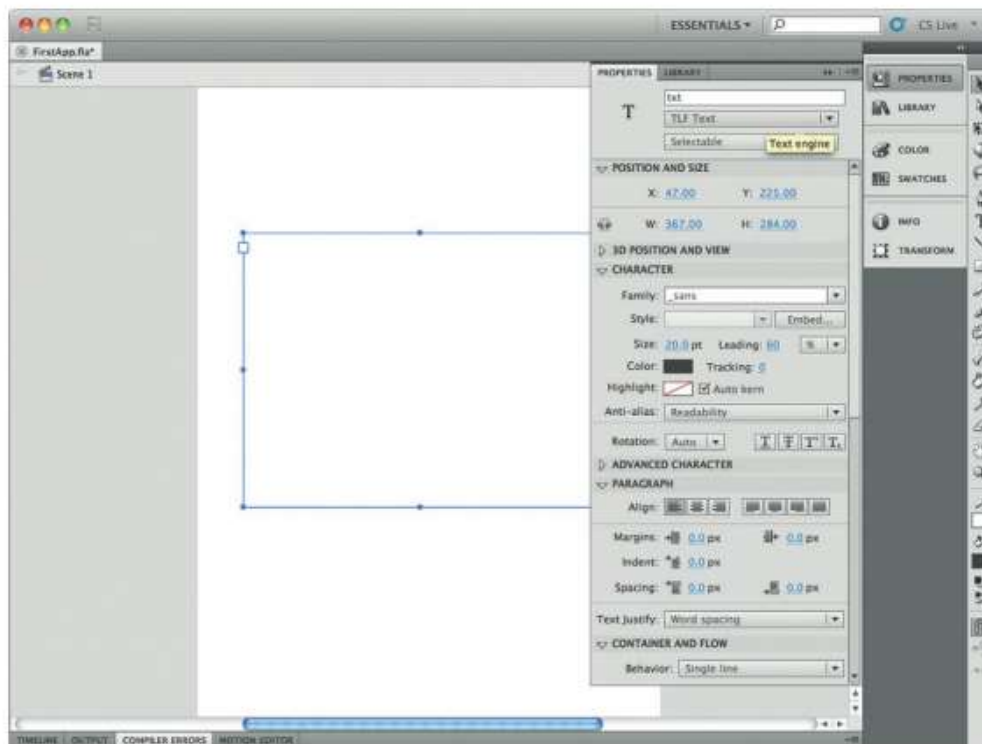
7. Change the font size to 20.

8. With the text field still selected, change the text type to Dynamic Text.

9. Give the text field an ID of txt.



*Figure 5: Currently there is only the AIR for Android template, but you can create your own for tablet devices.*



*Figure 6: Use text tools that you are already familiar with in Flash*

10. Open the Actions window and add the following ActionScript. The goal for this is to show you that the ActionScript you have been using all along will work. Enter the following ActionScript (Figure 7): `txt.text = "hello, world";`

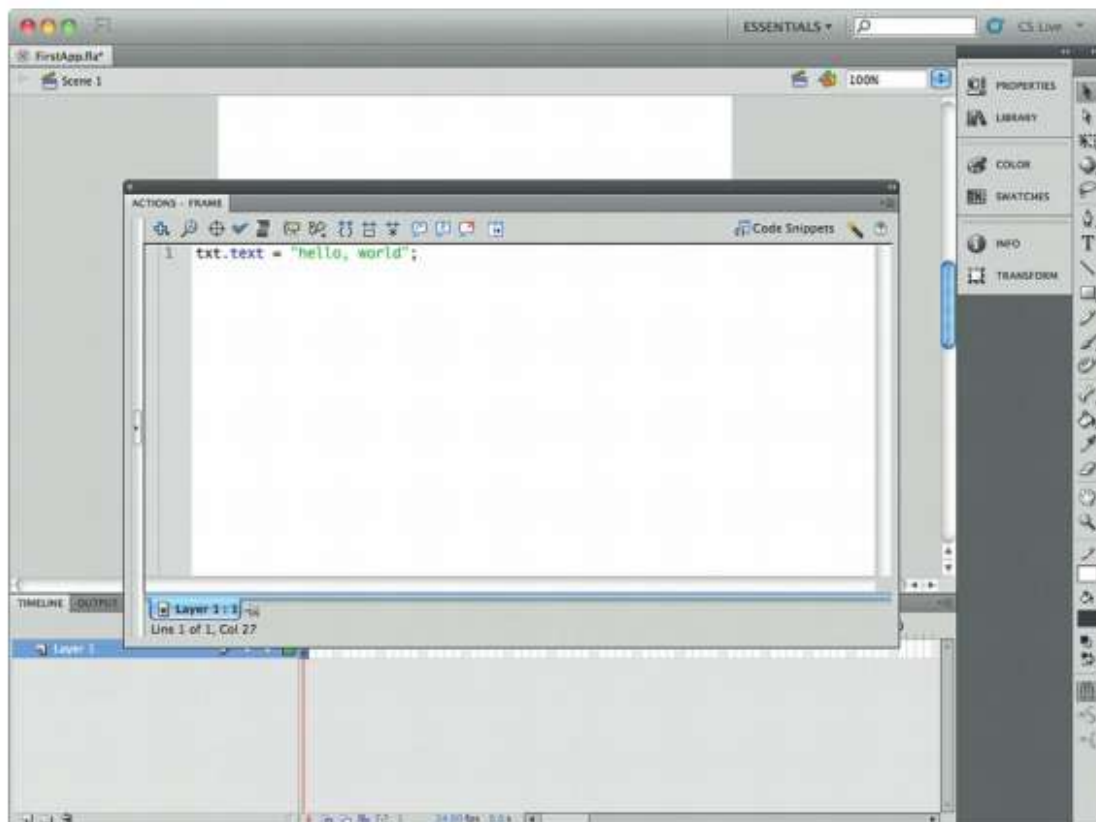
11. At this point you can test your Flash movie by pressing CTRL+ENTER. The movie should show you the text "hello, world" on your screen.

12. The next steps are to convert the Flash movie into an Android application.

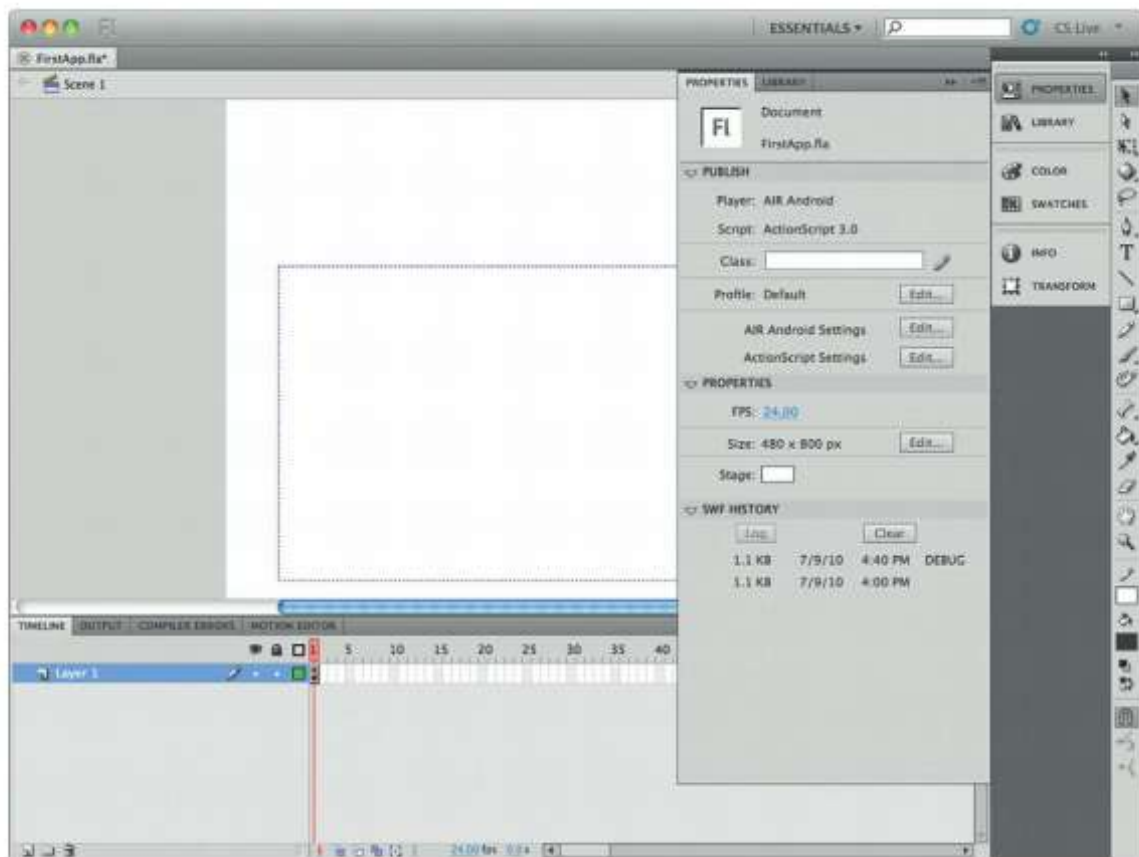
13. Select the Stage and choose the Properties panel. In the Profile section you will see AIR for Android Settings. Select the Edit... button (Figure 8).

14. The Application and Installer options window will open. Across the top of the window you will see three buttons, General, Deployment, and Icons, that toggle three different settings windows (Figure 9).

15. The General button shows you the following settings:



*Figure 7: Add dynamic content using ActionScript.*



**Figure 8: The Android application is created using special publish settings.**

- Output file
- App name
- App ID
- Version
- Aspect ratio
- Full screen
- Auto orientation
- Included files

16. The Output file is the location of the final file that will be installed on your android device.

The file format for Android apps is APK.

For this example you can keep the default filename. It should be called FirstApp.apk, and will save to the same folder as your Flash FLA file.

17. The App name is the name of the app as it will appear on the Android phone. The default is to use the name of the FLA file. Change the name to My First App.

18. The App ID is used when you publish your app to the Marketplace. For now you can keep the default, First App.

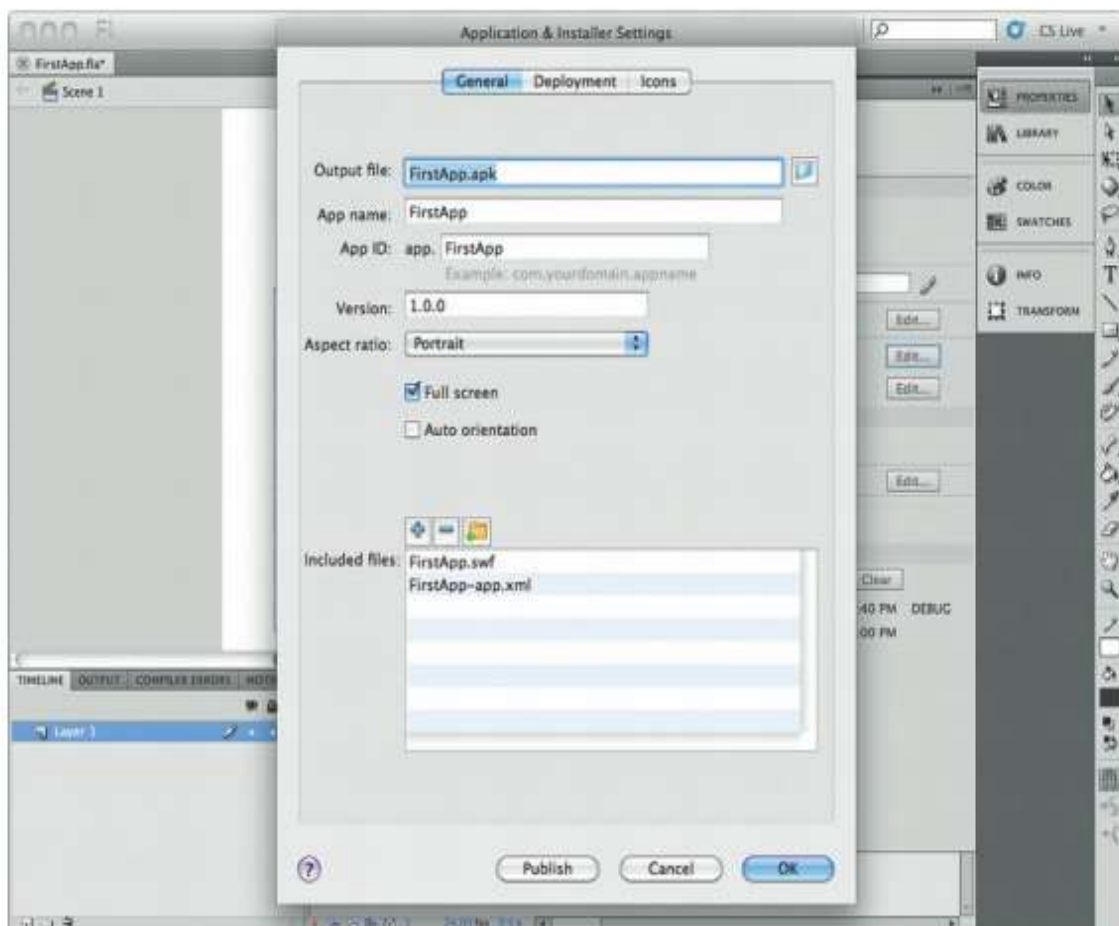
19. The version number allows you to add a version number to your Android app. It is up to you how you want to number your versions.

20. The Aspect ratio forces the default presentation of your Android app into either Landscape or Portrait. For now, keep the Aspect ratio as Portrait. Later, when you develop your first games, you

will learn how to design for Landscape aspect ratio.

21. Select the checkbox for Full screen. The Full screen setting forces the application to use up the whole screen and hide the status bar on the Android phone.

22. Do not select the Auto orientation checkbox. Auto orientation will allow the app to rotate as you rotate your phone.



**Figure 9:** You need to modify three screens to create your Android apps.

23. The Included files section allows you to add additional files into your final APK package. This can include files such as video, audio, and other SWF movies. You do not need to worry about that at this time.

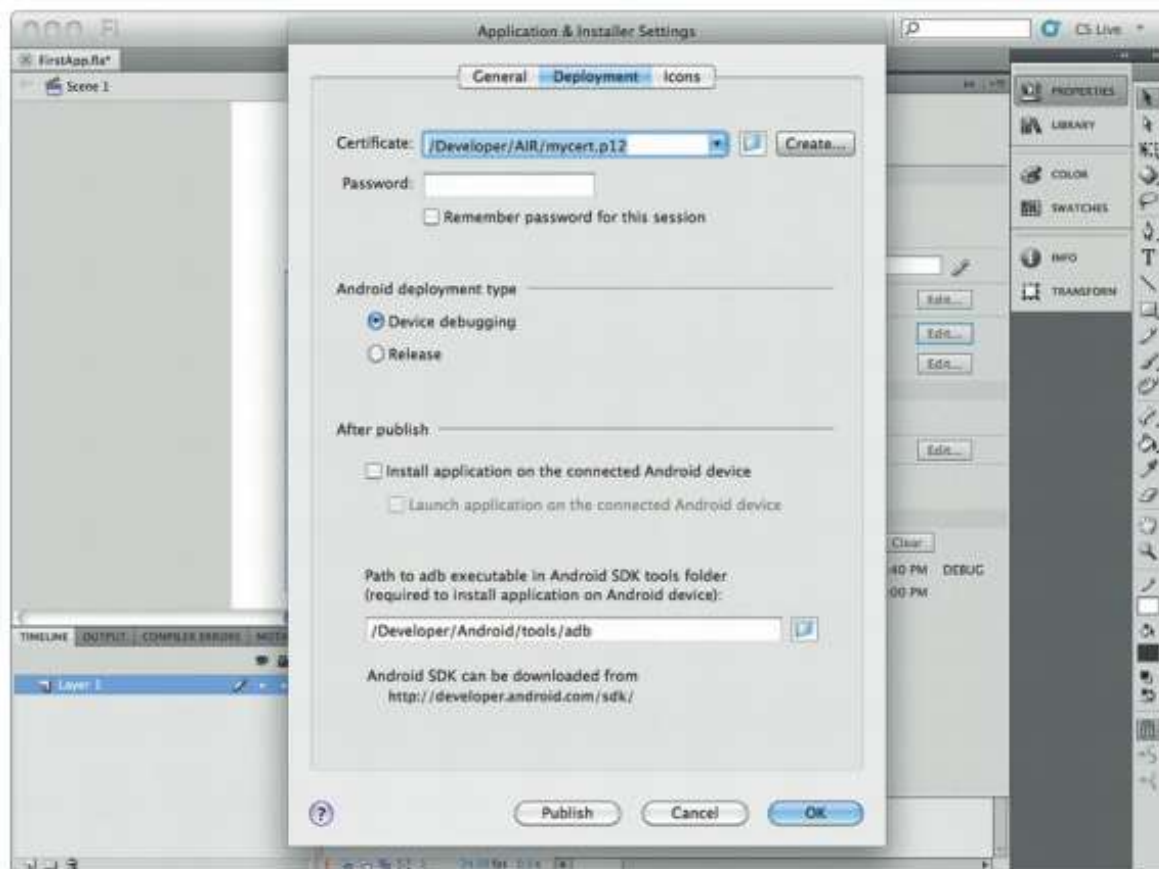
24. Now, select the Deployment button to go to Deployment screen (Figure 10). Each AIR app you build for Android requires a certificate. For development purposes you can use the same certificate

over and over. Let's create a Developer certificate.

25. Select the Create button. A new screen will open, asking you for additional information for the certificate.

26. For Publisher Name, Organization Unit, and Organization Name, insert Self. This is not a magical term, you can really enter anything you want.

27. Select the country from the drop-down menu.



**Figure 10: The Deployment tab controls how you build your application for Android.**

28. Enter a password. Make sure you remember the password because you will need to use it for future applications.

29. You can use the default 1024-RSA certificate strength.

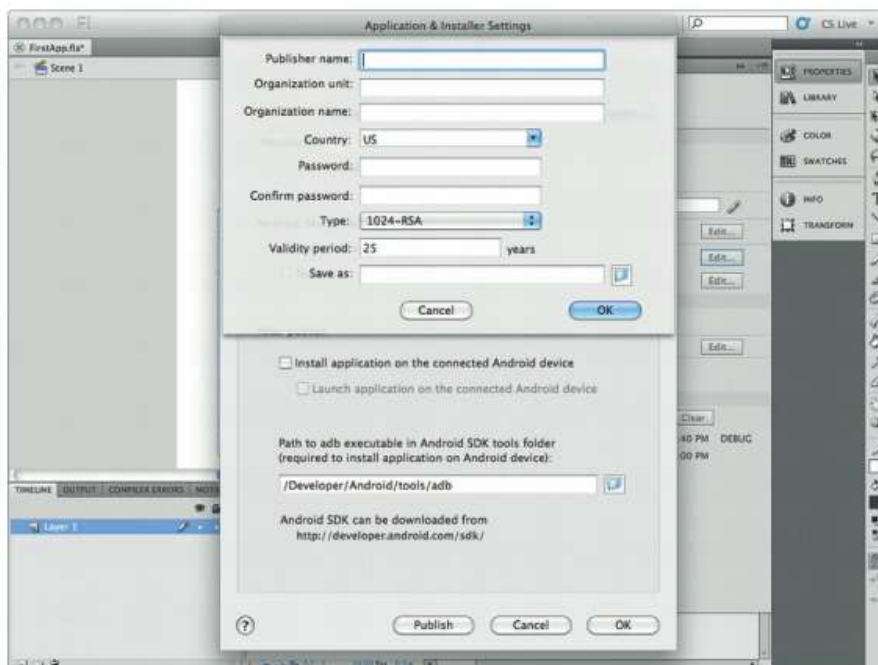
30. The default validity period is 25 years (Figure 11). That should be good enough for what we are doing.

31. Select the folder where you would like to store the certificate. The certificate will default to the file name mycert.p12.

32. Select OK. A window will pop up stating that a “Self signed certificate has been generated.” Select OK.

33. You will go back to the Deployment window. Enter your password. Choose the Remember password for this session checkbox. While you have this FLA file open, you will not need to keep re-adding the password each time you compile the file.

34. The Android deployment type option allows you to choose Device debugging or Release. For now, select the Device debugging option.



*Figure 11: A certificate is valid for 25 years*

35. Flash can install the final APK file directly onto your Android device for you. This is where you need to have the downloaded Android SDK. The After publish section will install the Application on your device but you need to have the Android SDK ADB tools. You can find the ADP tools within the Android SDK's Tools folder.

36. Don't worry about icons at this time.

37. Select the Publish button.

38. The app is small and should take only about 15 seconds to publish. You have now created an APK file and, if you selected the install options, you now have your first Android app running on your phone. How cool is that? Knuckle punch!

## CONCLUSION

At this point you have your first application running on your Android phone. The good news is, now that you have one application running you do not need to go through the hard work of installing JRE, Android SDK, AIR for Android, or a Developer's certificate again. You have done the hard work. Now you can focus on creating great AIR solutions with Flash CS5 for the Android platform.

## REFERENCES

1. Flash Mobile Developing Android & iOS Applications (2011), pages 7 to 25, by Matthew David
2. GAME AND GRAPHICS PROGRAMMING FOR IOS AND ANDROID® WITH OPENGL® ES 2.0, pages 2 to 8, by Romain Marucchi-Foino
3. Android™ Open Accessory Programming with Arduino™, pages 4 to 10, by Andreas Göransson and David Cuartielles Ruiz
4. Beginning Android Games, By: Mario Zechner
5. Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers), By: Ed Burnette
6. Android Application Testing Guide, by Diego Torres Milano
7. Android vs. iOS: Comparing the Development Process of the GQueues Mobile Apps, 1st ed.

2011 Edition, by Jeff Friesen and  
Dave Smith

8. Learning Mobile App  
Development, A Hands-on Guide  
to Building Apps with iOS and  
Android, pages 13 to 26, by Jakob  
Iversen and Michael Eierman
  
9. Android Programming: The Big  
Nerd Ranch Guide (Big Nerd  
Ranch Guides), By: Bill Philips &  
Brian Hardy