

AIR-Adobe Integrated Runtime

S.Sangeetha

ABSTRACT: AIR stands for Adobe Integrated Runtime. A runtime, like AIR, is a type of application or collection of application resources that you install on your computer for other applications to use in order to run. These applications can be either HTML/JavaScript based or SWF (ActionScript 3) based. When played in AIR, each has access to unique APIs for doing a lot more than what is capable in the browser.

What is AIR?

AIR (code name Apollo) is a new technology by Adobe allowing web developers create Rich Internet Applications (RIAs) for the desktop. As desktop applications, your RIAs can take advantage of additional features such as access to the file system, control over application windows, and support for drag and drop as well as everything they could when played in the browser.

AIR stands for Adobe Integrated Runtime. A runtime, like AIR, is a type of application or collection of application resources that you install on your computer for other applications to use in order to run. One of the main advantages of a runtime is that with runtimes installed on multiple platforms (Windows, OS X, or Linux) you can develop an application once without worrying about platform-specific programming. All of that is handled behind the scenes by the runtime. Another advantage is that applications built for the runtime will not have to bear all of the basic, redundant application weight that they would otherwise have to include in order to run by themselves. Instead, all of the common application data (which, in this case, is mostly the Flash Player and the WebKit web browser) is installed with the runtime once and not necessary for individual applications keeping them small and lightweight.

If you do not already have the AIR runtime installed, you can do so here:

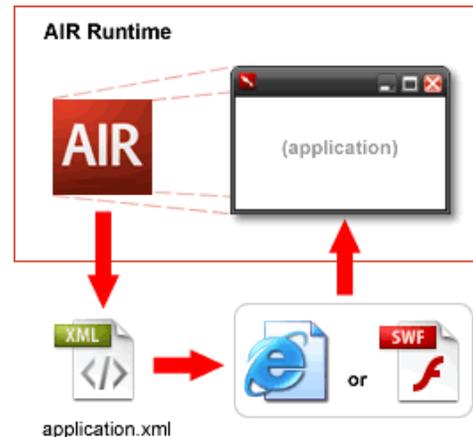
<http://labs.adobe.com/downloads/air.html>

AIR Applications

AIR "applications" themselves are really nothing more than one or more HTML or Flash SWF files (along with any other resource files they might use). AIR creates a container for running these files providing them with additional features that would otherwise not be available when run in the browser.

To be launched in AIR, an HTML or SWF application requires an application XML file (a.k.a. application descriptor file). You can almost look at this application XML file as the actual AIR application file. It is what is initially opened and read by AIR specifying application properties such as application window name and size

and identifying which file (HTML or SWF) is to be loaded into the AIR container. An AIR application cannot run without this XML file.



Installed Applications

AIR applications are usually distributed as single-file installers known as AIR files (.air). When opened, they copy just about everything that the AIR application needs to run on the user's computer including the HTML or SWF file(s), the application XML file for that application, and a small executable launcher to open AIR with the application XML. The only thing they do not include is the AIR runtime which is assumed to already be installed on the user's computer. If the AIR runtime is not installed, the user will be prompted to do so.

Once installed, AIR applications behave just like any other normal installed application. You get an icon in your start menu, a link on the desktop, and an application which, when launched, often looks like any other application would. Aside from the installer, users may not know the difference between an AIR application and any other application.

AIR SDK – Making Your Own AIR Applications:

SDK stands for Source Development Kit. SDKs provide developers with ways to create content for a certain framework or runtime etc. Adobe has released an SDK necessary for developers to create AIR applications. The AIR SDK consists primarily of:

- The AIR runtime
- A sample application XML file
- An application (adt) for creating AIR (.air) installers
- An application (adl) for running AIR applications without installing them

ADL stands for AIR Debug Launcher. It opens and runs application XML files. Being a debug launcher, it also provides developers with error messages that would otherwise be hidden from view when applications are run in AIR normally. This executable is a key component of the SDK as you, the AIR developer, will use it to run and test AIR applications before converting them into installable AIR files using ADT - the AIR packaging tool.

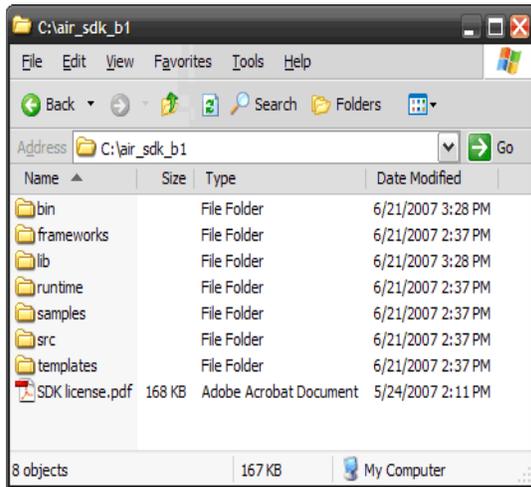
Hello World in AIR

The infamous Hello World application is usually the first step for anyone learning a new language. The goal: see the text "Hello World!" on the screen. This will cover how to get a Hello World using AIR.

What you will need:

- The Adobe AIR SDK
- A Text Editor

If you haven't already, download the [Adobe AIR SDK](#). The SDK itself does not need to be installed. It's just a collection of files to help you get started using and developing for AIR. After you download it, extract the files to a location on your hard drive. Many people like putting SDK's in their root directory on their file system. The path taken will be C:\air_sdk_b1\ though you can place them anywhere.



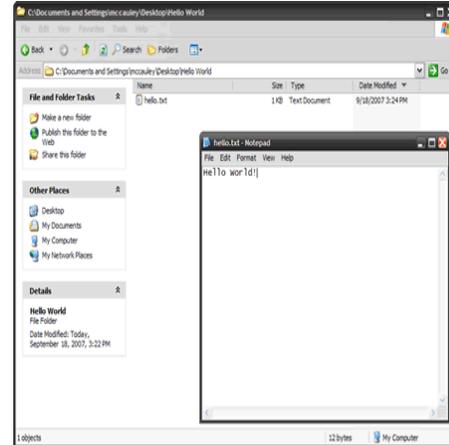
Note that the SDK (and the AIR runtime) is currently in beta. This means its a pre-release version of the software which may still have bugs and other issues which will be addressed in a later version. As you make applications using the beta SDK, keep in mind that you will need to update these applications when the full, final version of AIR is released. But don't worry; this process should be mostly painless.

Creating your application content:

1. Create a new folder somewhere on your hard drive to keep your Hello World AIR application files. I will be using a folder on my Desktop called Hello World
2. Using a text editor, create a document with the text:

Hello World!

3. Save this document as hello.txt

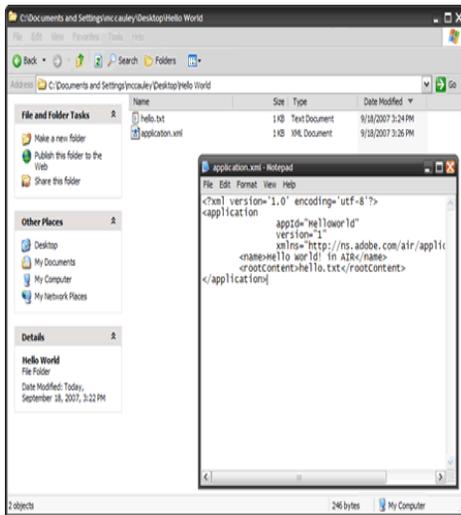


This file will represent your application content. Normally this would be an HTML file or a Flash SWF file, but for simplicity here, we're just using a text file, and since Webkit, the HTML renderer used by AIR (as well as just about all browsers), can render text files as HTML, it will be fine for now. All that remains is the application XML file.

Creating the application XML:

1. Using your text editor, create another document with the text:

2. `<?xml version='1.0' encoding='utf-8'?>`
3. `<application`
4. `appId="HelloWorld"`
5. `version="1"`
6. `xmlns="http://ns.adobe.com/air/applicati`
7. `on/1.0.M4">`
8. `<name>Hello World! in AIR</name>`
9. `<rootContent>hello.txt</rootContent>`
10. `</application>`
10. Save this document as application.xml



This file is the XML file that is used to describe your application. This particular file has the bare minimum necessary to run an application with AIR. Usually an AIR application's XML file would contain a lot more, but for now, this is all that is needed to get rolling. Lets take a look at what it consists of.

```
<application
    appId="HelloWorld"
    version="1"

    xmlns="http://ns.adobe.com/air/applicati
on/1.0.M4">
```

This is the root tag of the XML document, the application tag. This is the container for all of the information relating to the AIR application. It's three required attributes are appId, version, and xmlns. appId - A unique identifier for an AIR application that separates it from other applications. Often a reverse domain naming convention is used to make sure the id is unique. So, for example, the owner of example.com might use "com.example.HelloWorld" as an appId value. version* - The version of this application. If a user attempts to install an application when it is already installed, this value will be checked to determine if the current installation is newer or older than the new. xmlns - xmlns attributes in XML define a namespace for XML nodes. Here, the application element and all of its child elements will use the namespace with the provided URI (http://ns.adobe.com/air/application/1.0.M4). This URI is unique to the version of the AIR runtime used to run this application.

* The version attribute is not required when running an AIR application through ADL but is required when packaging with ADT.

The two required elements within the application tag included name and rootContent.

```
<name>Hello World! in AIR</name>
```

The name element in the application XML file provides the application with a name. This name is used by the operating system to name your window in areas like the task switcher, the task bar, and the application window title. For this Hello World AIR application, you will see "Hello World! in AIR" in the application's title bar when it launches.

```
<rootContent>hello.txt</rootContent>
```

The rootContent element references the application content file. This will need to be relative to the application XML file (you can use just the file name or a path to that file) and must be a file that can be opened natively in Webkit or Flash; this usually means an HTML or SWF file. Here a text file is used but works since it can be read as HTML and since this application serves no other purpose other than to just display text.

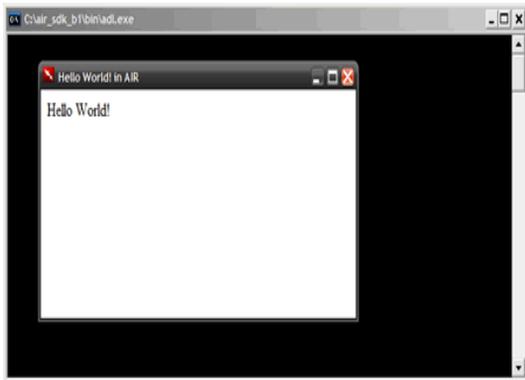
Though it is not necessary here to name the application XML file application.xml, when packaged with ADT, it will be renamed to application.xml and saved as such when installed on a user's system. Keeping that naming convention can more easily help identify this file as being the application XML file for this project. However, if you are using the same directory to store multiple AIR applications, each will need to have uniquely named application XML files.

Using the AIR SDK with ADL, packaging and installation is not needed to run your application. All you need to do is use ADL to launch your application XML.

Running your application:

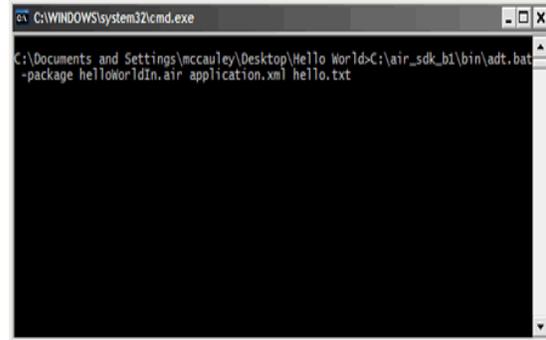
1. With a new Explorer or Finder window, navigate to the bin folder in your AIR SDK. It should contain the adl.exe and adt.bat files.
2. From directory in which you saved the application XML file for the Hello World application, drag and drop application.xml onto adl.exe

That's it. A command window will appear shortly followed by an AIR application window with the text "Hello World!" You have just created your first AIR application!



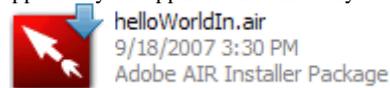
The full path look something like:

```
C:\air_sdk_b1\bin\adt.bat -package
helloWorldIn.air application.xml hello.txt
```



5. Hit enter

ADT will create a file called `helloWorldIn.air` that should appear in your application's directory.



Double-click on that file and you will be prompted to install Hello World! in AIR.

Packaging AIR Applications

Once you have a completed AIR application running through ADL, the next step is to package into an AIR file and distribute it to users. This is handled by ADT, the AIR packaging tool.

ADT is a command-line tool (written in Java, so it requires the [Java runtime](#) to be installed on your computer), so it's a little more difficult to use. Instead of being able to drag and drop like you can with ADL, you will need to run the application through a Command Shell or Terminal window. There you would run ADT with command-line arguments specifying the name of the AIR file to create and what files to include. Its format is

```
adt -package <name of .air file to create> <location
of application XML> <files to include>
```

Packaging your application

1. Open a command line window
2. Navigate to the location of the Hello World application. An easy way to do this is to type `cd` (followed by a space) in the command line and then drag the desired folder directly from Explorer or Finder into the command line window. Its path will automatically be added. Hit enter to navigate to that location.
3. Type the path to `adt` in the `bin` folder of the SDK into the command line (you can use drag and drop with ADT to quickly add it there as well)
4. After the path on the same type:

```
-package helloWorldIn.air application.xml
hello.txt
```

Moving Forward With AIR:

Congratulations! You have just successfully created and packaged your first AIR application! Though trivial in purpose, it served as a good demonstration of what it takes to get an AIR application up and running. Next you will need to build an application that serves a better purpose and makes better use AIR's features.

Before you go off making AIR applications left and right, however, you should stop to consider whether AIR is necessary for what you are trying to accomplish. The basic premise of AIR is to give established (or aspiring) web developers an easy platform for developing desktop applications. If you find yourself not using any of the features unique to the AIR runtime, perhaps AIR isn't necessary for your application. After all, why require that a user go through the AIR installation process if all they would need to do is run your application through their browser. Don't forget too that users would need to have the AIR runtime installed in order to run AIR applications - a dependency that can't always be taken lightly. AIR should only be used if you're really taking advantage of what AIR has to offer and feel the requirements for deployment are acceptable.

If you find that AIR is what you need, you can begin to develop applications that make much better use of the runtime. These applications can be either HTML/JavaScript based or SWF (ActionScript 3) based. When played in AIR, each have access to unique APIs for doing a lot more than what is capable in the browser. This includes:

- File I/O API
- SQLite embedded database
- All functionality within Flash Player 9, including complete network stack
- Windowing APIs
- Command-line tools (ADL and ADT)
- HTML within Flash content
- Top-level HTML applications
- ActionScript / JavaScript Script bridging
- Flex Builder and Flex Framework support for authoring AIR application
- Application command-line arguments
- Drag and Drop Support
- Rich Clipboard Access
- Native Menu API (Mac only in beta)
- Service Connectivity API
- File Type Association
- Application icons
- PDF Support
- ([source](#))

When making SWF based applications, you'll find that much of the manual labor creating application XML files and AIR packaging is taken care of through [Flex Builder 3](#) and [Flash CS3](#) using the [AIR update for Flash](#). Those tools allow you to create AIR-specific documents that use ADL and ADT behind the scenes to make AIR application developer much more seamless.

Similarly, [Dreamweaver CS3](#) has an [AIR update for Dreamweaver](#) that does this for HTML/JavaScript-based AIR applications when developed in Dreamweaver. This doesn't mean these tools are necessary for AIR applications, however. We've already demonstrated that an AIR application can be developed with the SDK and a text editor alone.

Conclusion:

In the end making an AIR application is not complicated. Once you understand the basics, working with AIR will become second nature.

References:

- 1 www.google.com
- 2 <http://labs.adobe.com>
- 3 <http://en.wikipedia.org>

Biography:



S.Sangeetha received her B.C.A degree in Kongu Arts & Science College affiliated to Bharathiyar University,Coimbatore. She has been studying MCA in K.S.R College of Engineering. Her area of interest includes Operating System and SQL. Email ID:ssangee007@gmail.com