Aviary Feather Android Setup Guide

Aviary Inc.

1. Introduction

This document will guide you through the creation of a sample application using the AviaryFeather Android library.

1.1 Prerequisites

I assume you already have the Android environment installed on your system and Eclipse with the required ADT plugin. See http://developer.android.com/sdk/installing.html and http://developer.android.com/sdk/eclipse-adt.html if you need instructions on how to setup the Android environment.

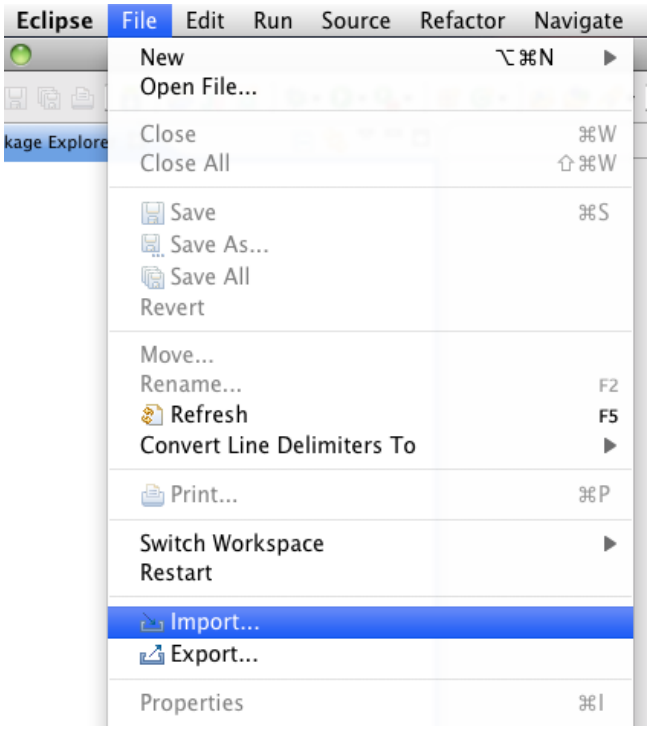
You will also need an Aviary API key/secret pair to access the remove effect API. To sign up or learn more, please visit http://developers.aviary.com/geteffectskey.

2. Workspace setup

First, we'll need to import the 2 Eclipse projects into our workspace.

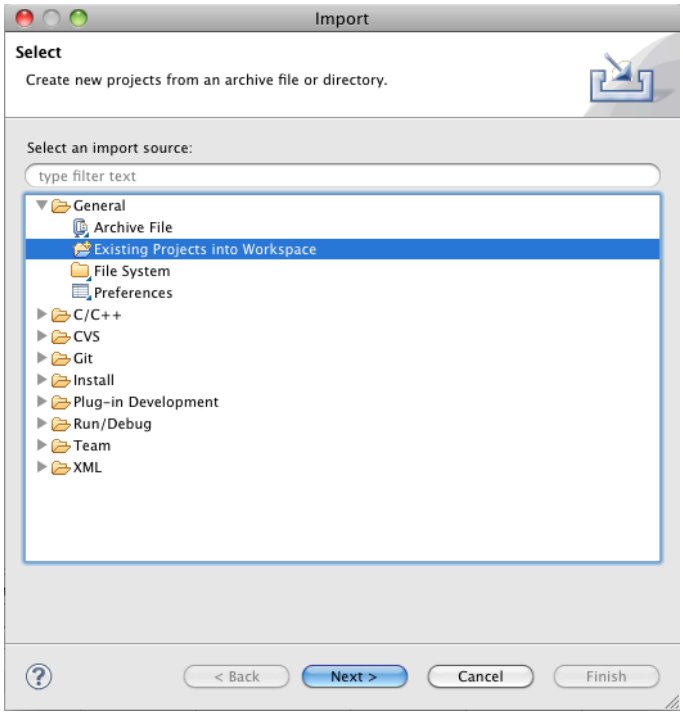
2.1 Import Project

Open Eclipse and select "Import" from the file menu.



2.2 Set Up Import

The import dialog will appear. From the list of import options, select “Existing Projects into Workspace,” and then click “Next.”

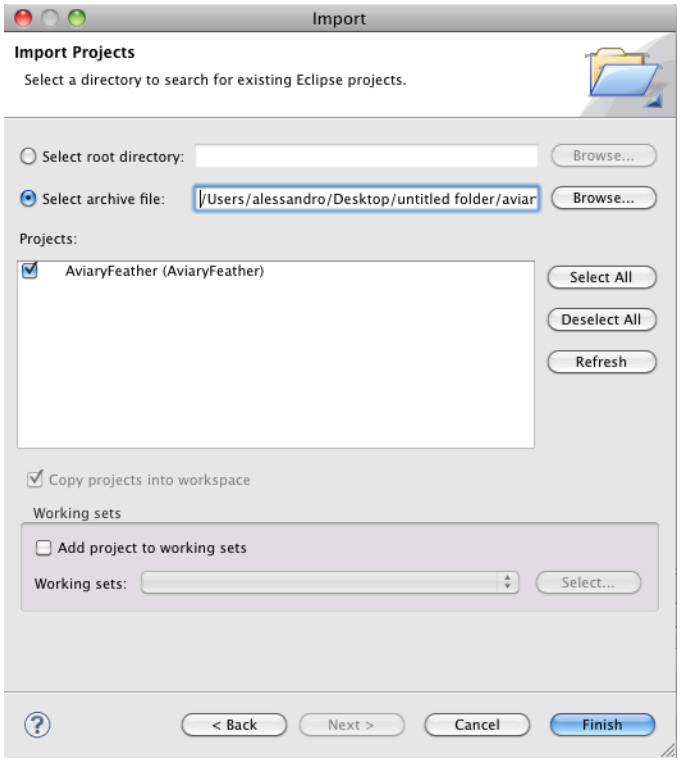


2.3 Select the File

In the new dialog, click on the “Select archive file” radio button and then click the “Browse” button on the right. From here, select the **aviaryfeather.zip** file included with this document.

Click on the “Finish” button at the bottom of the dialog. A new Android library project called “AviaryFeather” will be created in your current workspace.

This is the required library project which you must include in your application if you want to use Feather to manipulate images.



3. Sample Application

Next, we need to create an Android application in order to use Feather.

You can see a real example of how to use Feather by opening the included **sample-app.zip** project.

Just import the sample application by following the same procedures described above, but select **sample-app.zip** at step 3. A new project called “AviaryLauncher” will be created in your workspace.

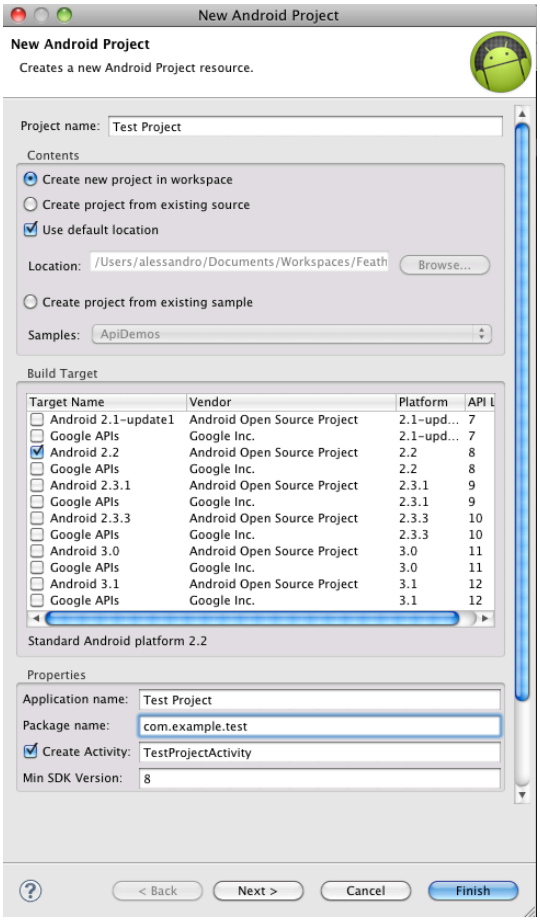
The imported application should have all the references already set and it should be ready to use. If you want to include AviaryFeather in a different Android project or add it to a new one, follow the instructions in step 4; otherwise you can skip to step 5.

4. Include AviaryFeather in a new Application

If you don't want to use the included sample application to test Feather, here's a step by step guide on how to include Feather in a new Android application.

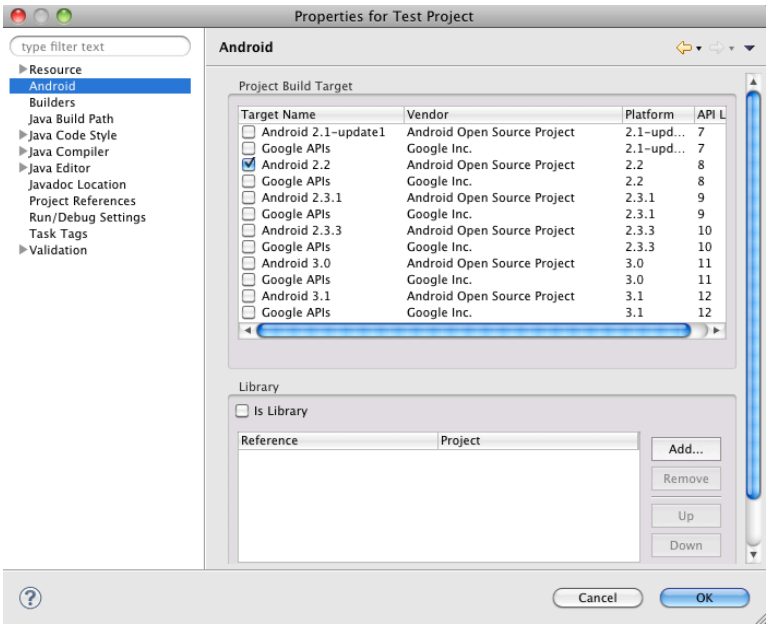
4.1 Create a new Android project

Just create a new Android project as usual from Eclipse.

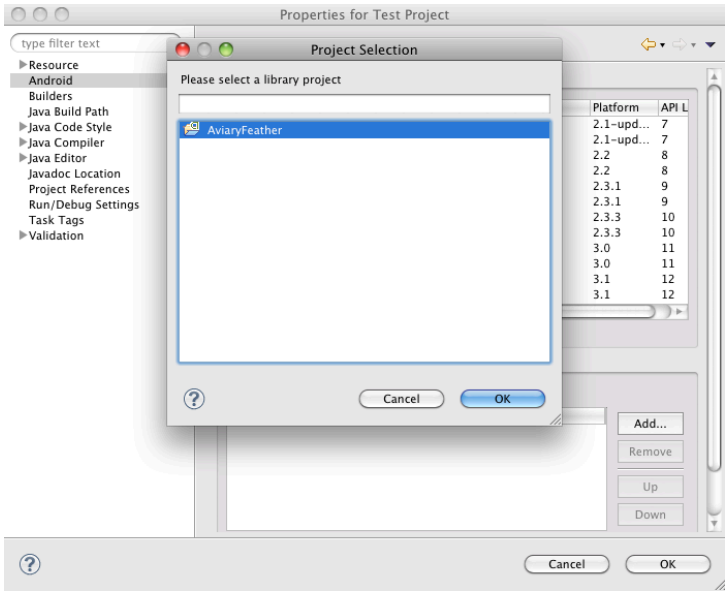


4.2 Project references

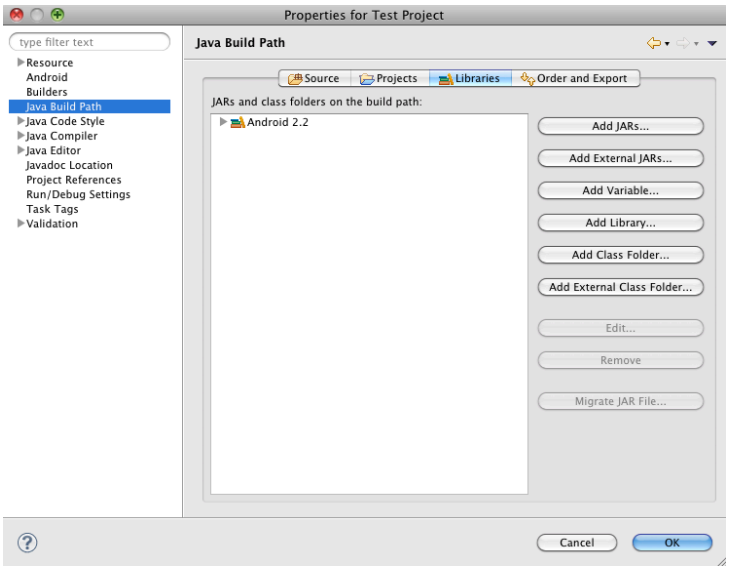
Once the new project has been created, open the project properties and navigate to the "Android" section.



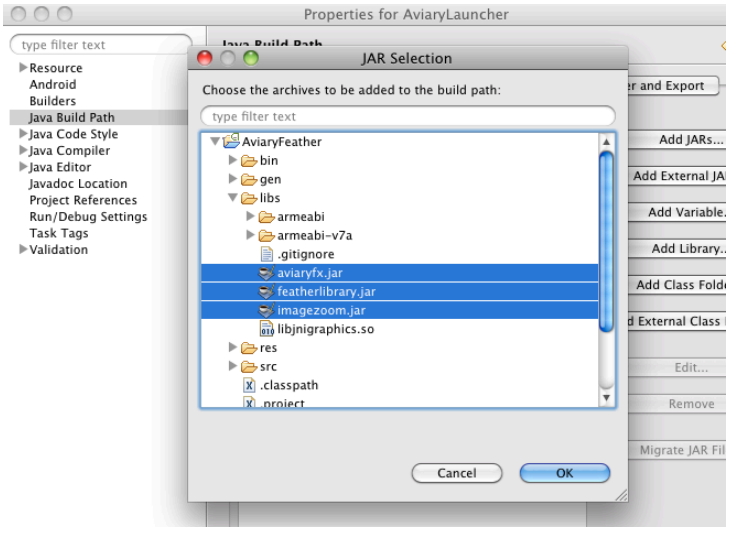
Click the “Add...” button of the “Library” subsection and select “AviaryFeather” from the dialog.



Next, navigate to the “Java Build Path” section of the project properties dialog and click on “Add JARs...” button of the “Libraries” subsection.



From here, select all the .jar files included in the “libs” folder of the AviaryFeather project (**aviaryfx.jar, imagezoom.jar** and **featherlibrary.jar**).



4.3 AndroidManifest.xml

Add some entries to the manifest file of your application.

4.3.1 Permissions

AviaryFeather requires both internet access and write access to external storage. To grant those permissions, add these entries inside the **AndroidManifest.xml <manifest>** tag:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

4.3.2 Activity declaration

Then, inside the **<application>** tag, add a reference to the FeatherActivity:

```
<activity android:name="com.aviary.android.feather.FeatherActivity"
       android:theme="@style/FeatherTheme.Custom"
       android:configChanges="orientation|keyboardHidden"
       android:screenOrientation="portrait" />
```

4.4 themes.xml

The **android:theme** entry in the manifest file is also required for Feather to work properly, so add an entry to your **themes.xml** file (if you don't have one, create a new file called **themes.xml** in your res/values folder):

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
       <style name="FeatherTheme.Custom" parent="FeatherTheme.Dark" />
</resources>
```

By default, this entry will use the default Feather theme.

If you'd like to customize the Feather UI, you can do that simply by adding entries to your "Feather.Custom" style. Check out the **themes.xml** file included in AviaryFeather/res/values for the list of available keys.

5. Invoke Feather

If you're calling Feather from a new application, you'll need to add the below code in order to start Feather. Otherwise (if you're using the demo application) you can find this code inside the ***MainActivity.java*** *file.*

In order to invoke Feather from your activity, you need to pass some parameters to the FeatherActivity. Here's an example of how to invoke the new activity:

```
// Create the intent needed to start feather
Intent newIntent = new Intent( this, FeatherActivity.class );

// set the source image uri
newIntent.setData( uri );

// pass the required api key/secret ( http://developers.aviary.com/geteffectskey )
newIntent.putExtra( "API_KEY", “xxx” );
newIntent.putExtra( "API_SECRET", “xxx” );

// pass the uri of the destination image file (optional)
// This will be the same uri you will receive in the onActivityResult
newIntent.putExtra( “output”, Uri.parse( "file://" + mOutputFile.getAbsolutePath() ) );

// format of the destination image (optional)
newIntent.putExtra( “output-format”, Bitmap.CompressFormat.JPEG.name() );

// output format quality (optional)
newIntent.putExtra( “output-quality”, 85 );

// you can force feather to display only a certain tools
// newIntent.putExtra( "tools-list", new String[]{"SHARPEN", "BRIGHTNESS" } );

// ..and start feather
startActivityForResult( newIntent, ACTION_REQUEST_FEATHER );
```

5.1 Intent parameters

Here's a description of the required parameters:

Uri (intent data)	This is the source uri of the image to be used as input by Feather
API_KEY/API_SECRET	api key and secret required to use remote filters. Go to http://developers.aviary.com/geteffectskey for more information on how to obtain your api key and secret
output	This is the uri of the destination file where Feather will write the result image
output-format	Format of the output file (jpg or png)
output-quality	Quality of the output image (required only if output-format is jpeg). 0 to 100
tools-list	If specified in the extras of the passed intent it will tell feather to display only certain tools. The value must be a String[] array and the available values are: SHARPEN, BRIGHTNESS, CONTRAST, SATURATION, ROTATE, FLIP, BLUR, EFFECTS, COLORS, RED_EYE, CROP, WHITEN, DRAWING, STICKERS

Note: If you're using our sample application, you only need to replace the api_key and api_secret constants inside the **MainActivity.java** file:

```
private static final String API_KEY = "xxxx";
private static final String API_SECRET = "xxxx";
```

5.2 Result parameters

Once the user clicks "save" in the Feather activity, the "onActivityResult" of your Activity will be invoked, passing back "ACTION_REQUEST_FEATHER" as requestCode.

The Uri data of the returned intent will be the output path of the result image:

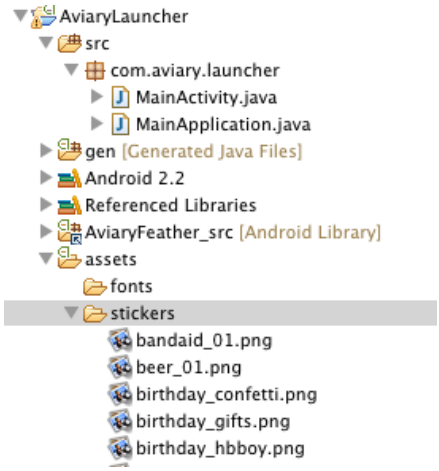
```
@Override
public void onActivityResult( int requestCode, int resultCode, Intent data ) {
	if( resultCode == RESULT_OK ) {
		switch( requestCode ) {
			case ACTION_REQUEST_FEATHER:
				Uri mImageUri = data.getData();
				break;
		}
	}
}
```

6. Extras

6.1 Stickers

Feather uses the “assets/stickers” folder of your application as input for the stickers tool, so your application must include the list of sticker files inside that folder. (The AviaryLauncher sample application already has a bunch of images inside its assets/stickers folder by default.)

Note: If your application does not have the above folder or if that folder is empty, the stickers tool will be automatically hidden in Feather.



6.2 Other configurations

Inside the AviaryFeather/res/values is a **config.xml** file. This file contains some application default values and can be modified before compilation.

6.3 UI Customization

Feather comes with the default **FeatherTheme.Dark** theme, the one you need to extends from your themes.xml file (see section 4.4). You can modify almost every part of the UI by overriding the values of FeatherTheme.Dark into your theme. Here some examples.
Open your res/values/themes.xml file which should be like this:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
	<style name="FeatherTheme.Custom" parent="FeatherTheme.Dark"></style>
</resources>
```

Now let’s say we want to change the top bar and bottom bar default font family. To do this just place a ttf font file into your “assets/fonts” directory (for instance “helvetica.ttf”) and add these lines inside the <style></style> tag of your themes.xml file:

```
<item name="toolbarFont">fonts/Helvetica.ttf</item>
<item name="bottombarFont">fonts/Helvetica.ttf</item>
```

Or let’s say you want to change the default toolbar background. Just add this line:

```
<item name="toolbarBackground">#FFCCCCCC</item>
```

For a complete list of customizables, just open AviaryFeather/res/values/themes.xml and see what’s inside the main <style> tag