

Introduction

The Android R1 Photo Effects SDK allows developers to include a suite of image effects in their applications.

The zip file includes the R1 Photo Effects SDK and an SDK demo app for installation reference.

Eclipse Demo Installation

1. Verify that the latest android tools have been installed in the SDK Manager.

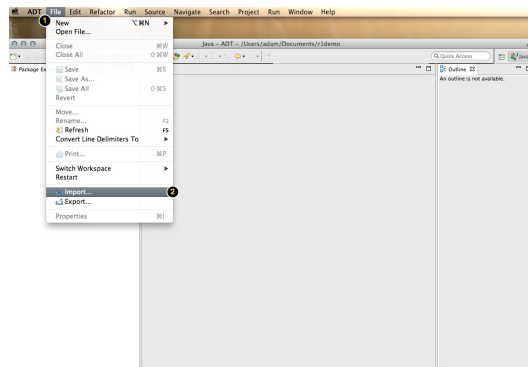
- The SDK Manager can be accessed through the window menu option in eclipse
- Installation requires version 22 of the Android SDK Tools
- Installation requires version 17 of the Android SDK Platform-tools
- Installation requires version 17 of the Android SDK Build-tools
- SDK requires Android version 2.3 to run

2. Verify that eclipse is up to date.

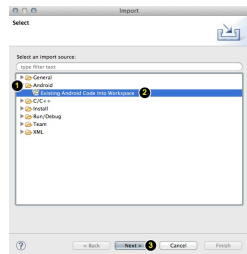
- Updates can be downloaded and installed from the help menu tab

3. Import the Photo Effects SDK.

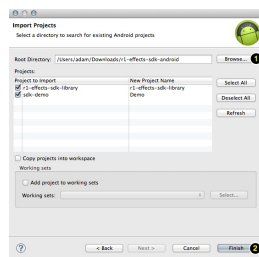
- Select File
- Select Import



- Select Android
- Select Existing Android Code Into Workspace
- Select Next

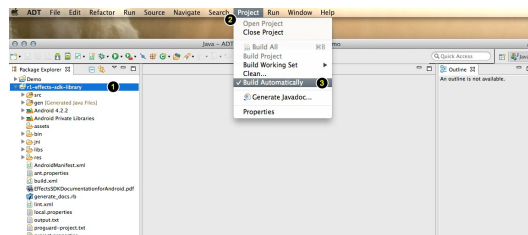


- Select Browse and locate the R1 Photo SDK [root] folder
- Verify that both r1-effects-sdk-library and sdk-demo are checked
- Select Finish



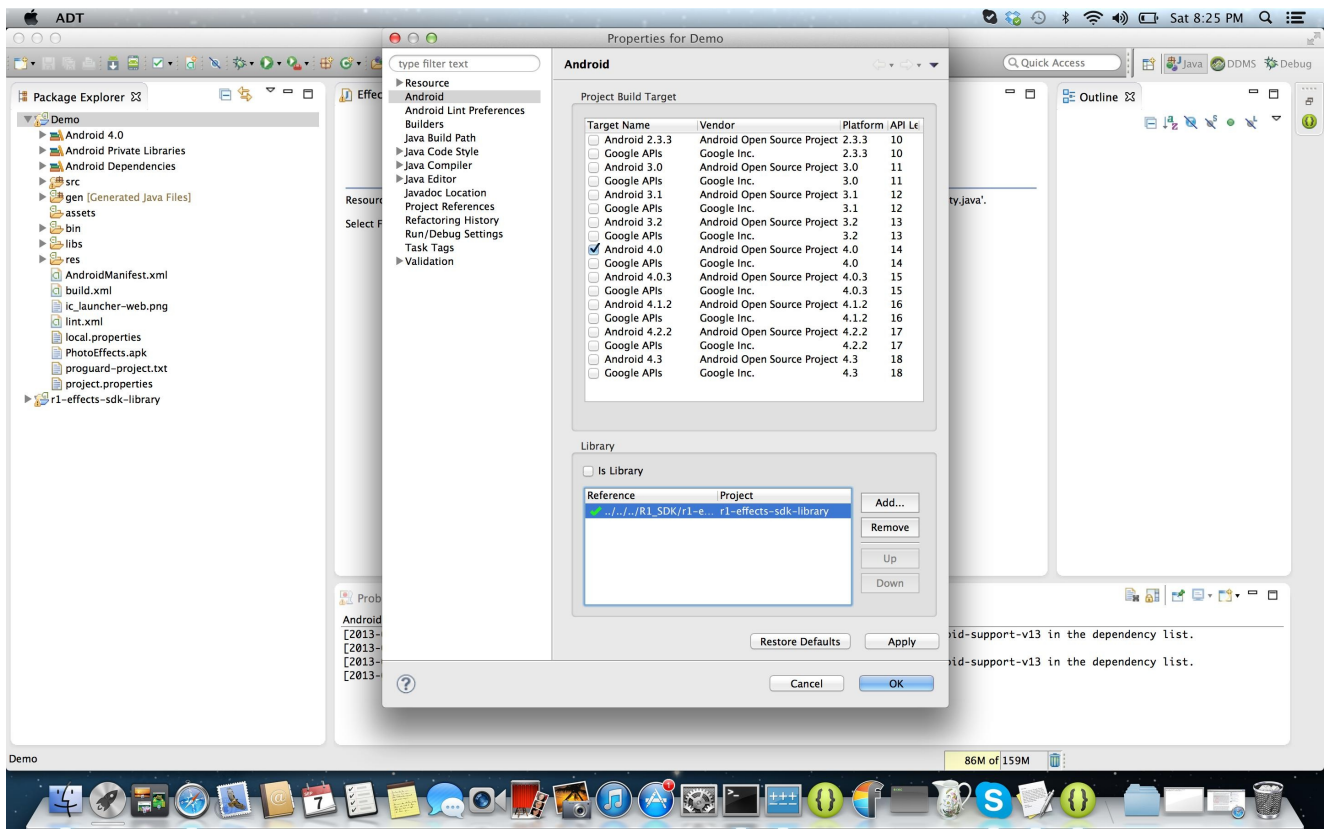
4. Disable automatic builds.

- Select the r1-effects-sdk-library project in the Package Explorer
- Select the Project menu tab
- Deselect Build Automatically



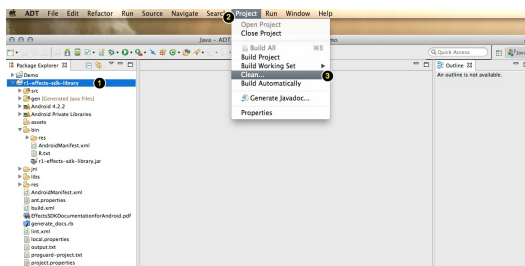
5. Add effects library to your Android project

- Right click on your project in the Package Explorer
- Select Properties
- Add r1-effects-sdk-library to your project. For the demo project it already been added.
- Select ok

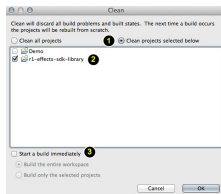


6. Clean the SDK Project.

- Select the r1-effects-sdk-library project in the Package Explorer
- Select the Project menu tab
- Select Clean

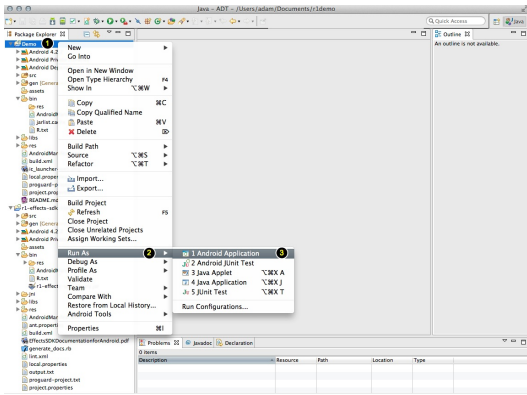


- Select Clean projects selected below
- Verify that only the SDK project is checked
- Deselect Start a build immediately
- Select OK



7. Run the Demo Project.

- Right-click the demo project in the Package Explorer
- Select Run As
- Select 1 Android Application
- The demo will build to a connected android device



Eclipse Custom Installation By Example

1. Configure the AndroidManifest.xml.

- Double click the AndroidManifest.xml file under the demo project in the Package Explorer
- Select the first tab to the right below the open window to access the XML.

```
// The following permissions must be inside of the manifest tag
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="com.android.messaging.AIRLOCK"/>
<uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>

// The following sdk reference must be inside of the manifest tag
<uses-sdk android:minSdkVersion="19" android:targetSdkVersion="19"/>

// The following feature must be inside of the manifest tag
<uses-feature android:glEsVersion="0x00000000" android:required="true"/>

// The following activities must be inside of the application tag
<activity android:name="com.radiomove.effects_sdk.CropImage" android:configChanges="orientation|keyboardHidden" android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

```

2. Initialize the SDK in activity onCreate (for the first activity loaded on application open).

- Inside of the onCreate function for the activity that will call the R1 Photo Effects SDK add the following line for SDK initialization
- The R1 Photo Effects app token and Google Play app signature are required.
- Google play app signature is required incase of in-app purchase of stock items. Empty string can be passed if developer doesn't want to support it.

```
R1PhotoEffectsSDK.getInstance().enable(getActivity().getApplicationContext(), "Your R1 SDK APP TOKEN", "Your App Signature");
```

3. Send an image to the R1 Photo SDK.

- Add the following to your fragment, where imagebitmap is the photo bitmap to pass to the SDK. Use getActivity().getContext() when running from the activity/fragment.

```
R1PhotoEffectsSDK r1sdk = R1PhotoEffectsSDK.getInstance();
r1sdk.launchPhotoEffects(
    getActivity().getApplicationContext(),
    imagebitmap,
    true,
    new R1PhotoEffectsSDK.PhotoEffectListener() {
        @Override
        public void onEffectComplete(Bitmap output) {
            if (null == output) {
                return;
            }
            // do something with output
        }
        @Override
        public void onEffectCancelled() {
            // user cancelled
        }
    }
);
```

Paid Effects Configuration

Developers can enable paid packs for their application through Google Play. To add paid packs, contact your account manager for more information. Paid effects will not be displayed unless users are signed in to the Google Play Store on their devices.

1. Sign in to the Google Play Developer Console.

- Login at <https://play.google.com/apps/publish>
- Select your application
- Select in-app Products from the left hand menu

2. Add each pack as an in-app product.

- Select the + Add new product button
- Select the Managed Product tab
- Copy/paste the product ID into the form and select Continue
- Copy/paste the product name into the title field
- Copy/paste the product description into the description field
- Set a price (we suggest 99 cents)
- Select the Auto-convert prices now button
- Select the Save button
- Select activate in the drop down menu to the right of the save button
- Repeat for each additional paid package