im360 Android SDK

Getting Started

Im360 SDK Android 4.0 and newer

Introduction

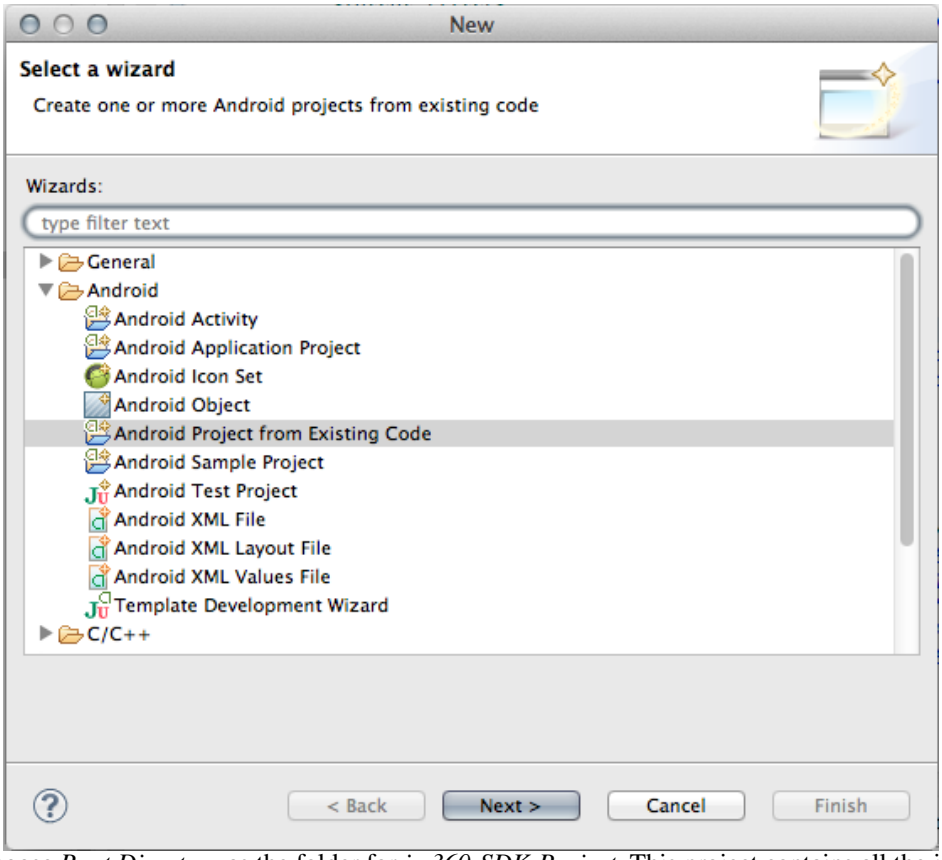
The im360 Android SDK is meant for use as a platform to create im360-Player-enabled Android apps, using Android Developer Tools (ADT). This document is designed to assist developers in the creation of such apps.

Requirements

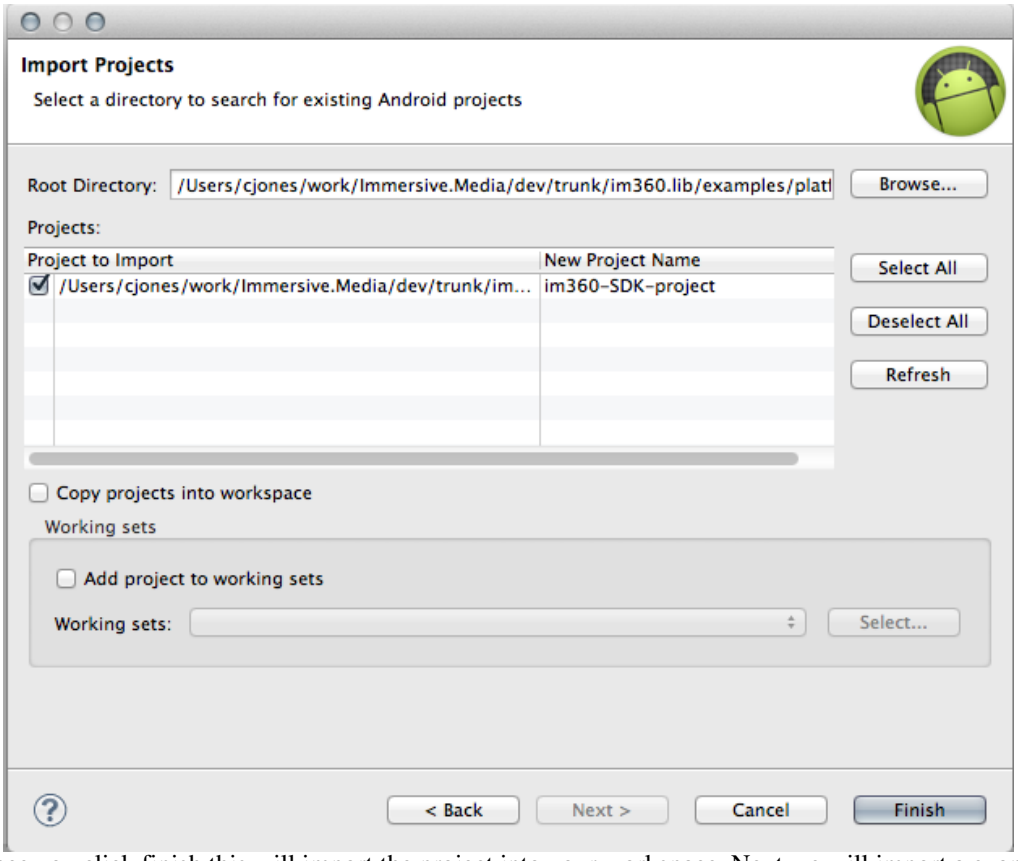
- im360 SDK Project for Android *(This folder will be supplied by Immersive Media)*
- im360 Android examples *(This folder will be supplied by Immersive Media)*
- Android 4.x Device (Important: The im360 Android SDK will not run in the Android Simulator. A real device must be used.)
- Mac/Windows/Linux running Android Developer Tools (Eclipse with the ADT Plugin) (At the time of this document version 22.3 is used)

Setting up the Android SDK Project and Example Project

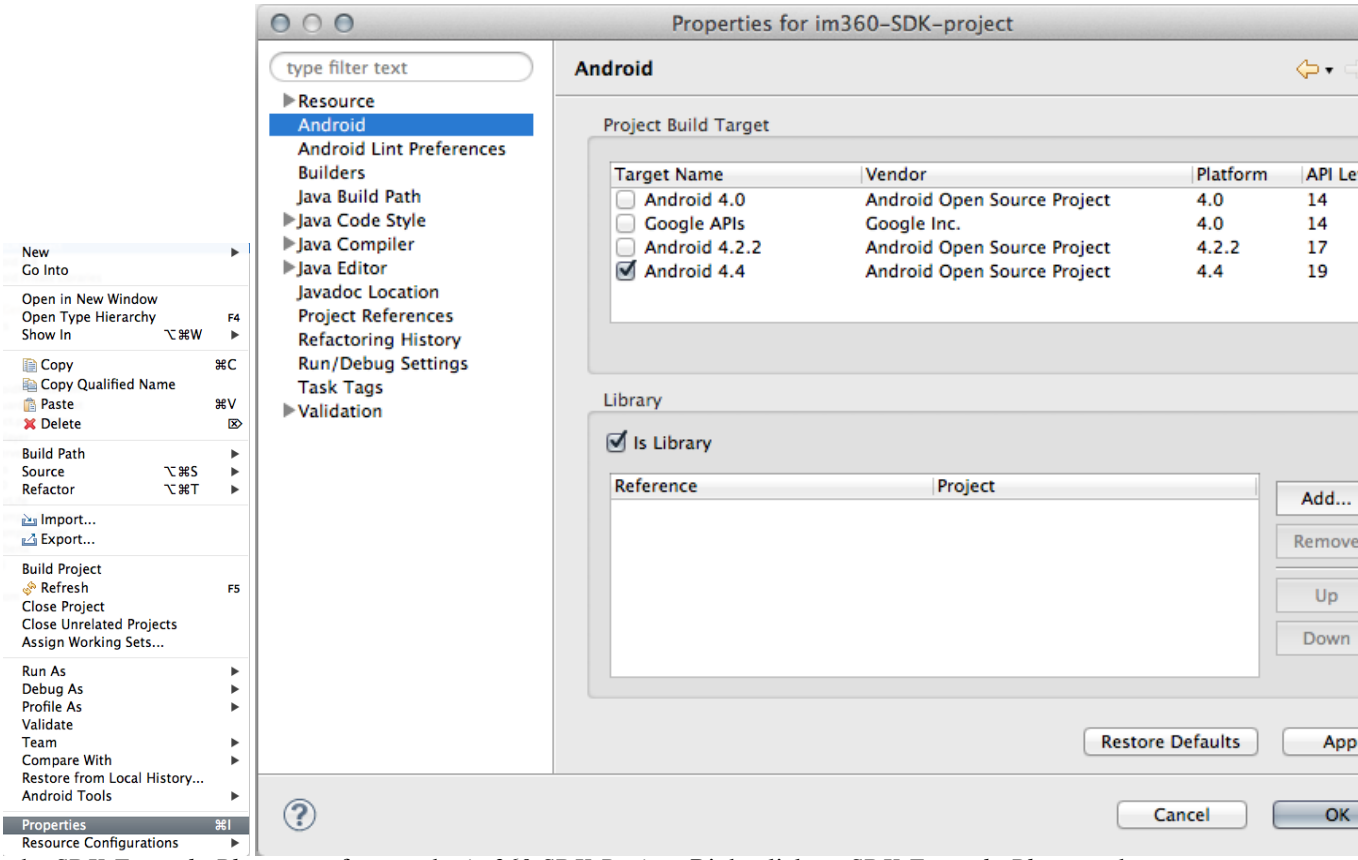
1. Create a new work space or use a existing one.
2. Then choose *File->New->Other* and choose "Android Project from Existing Code" and click "Next"



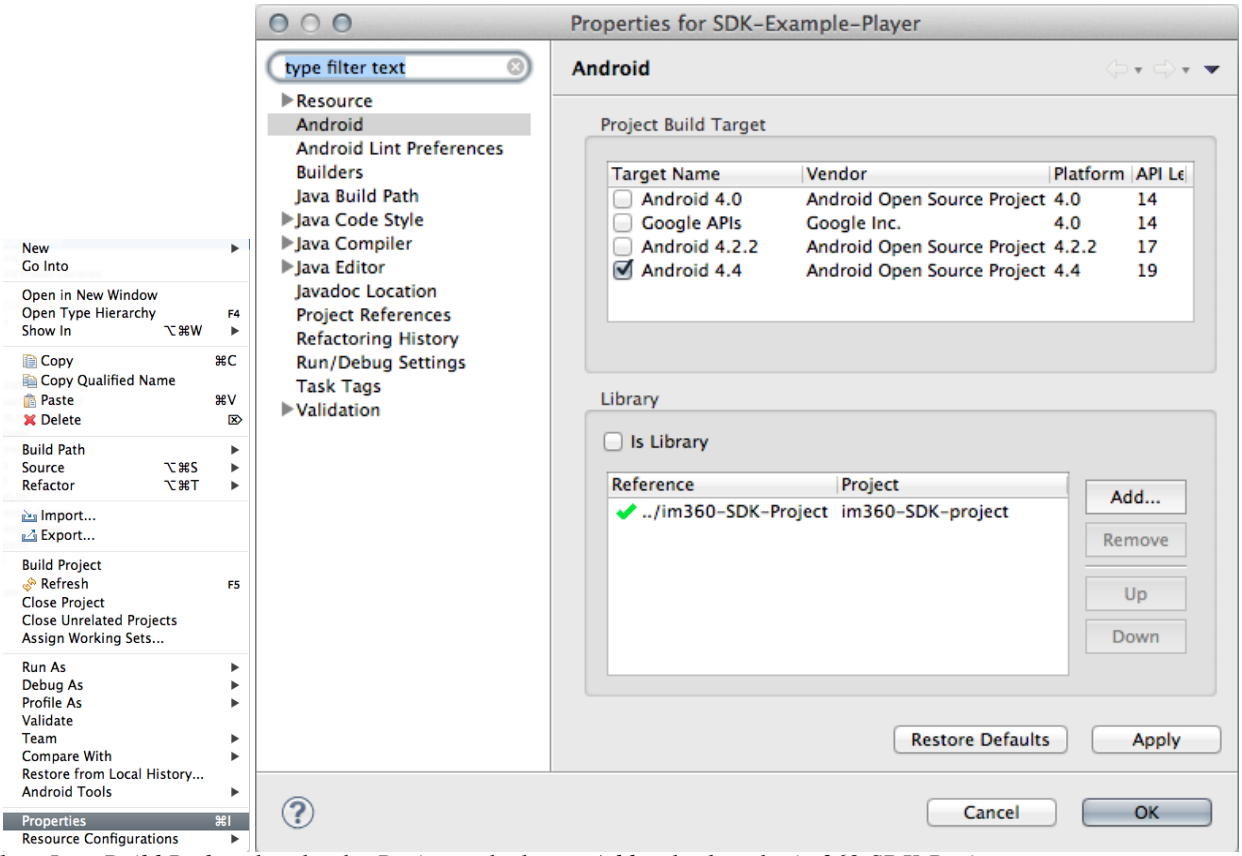
3. Choose *Root Directory* as the folder for *im360-SDK-Project*. This project contains all the jars and libs needs for the im360 sdk.



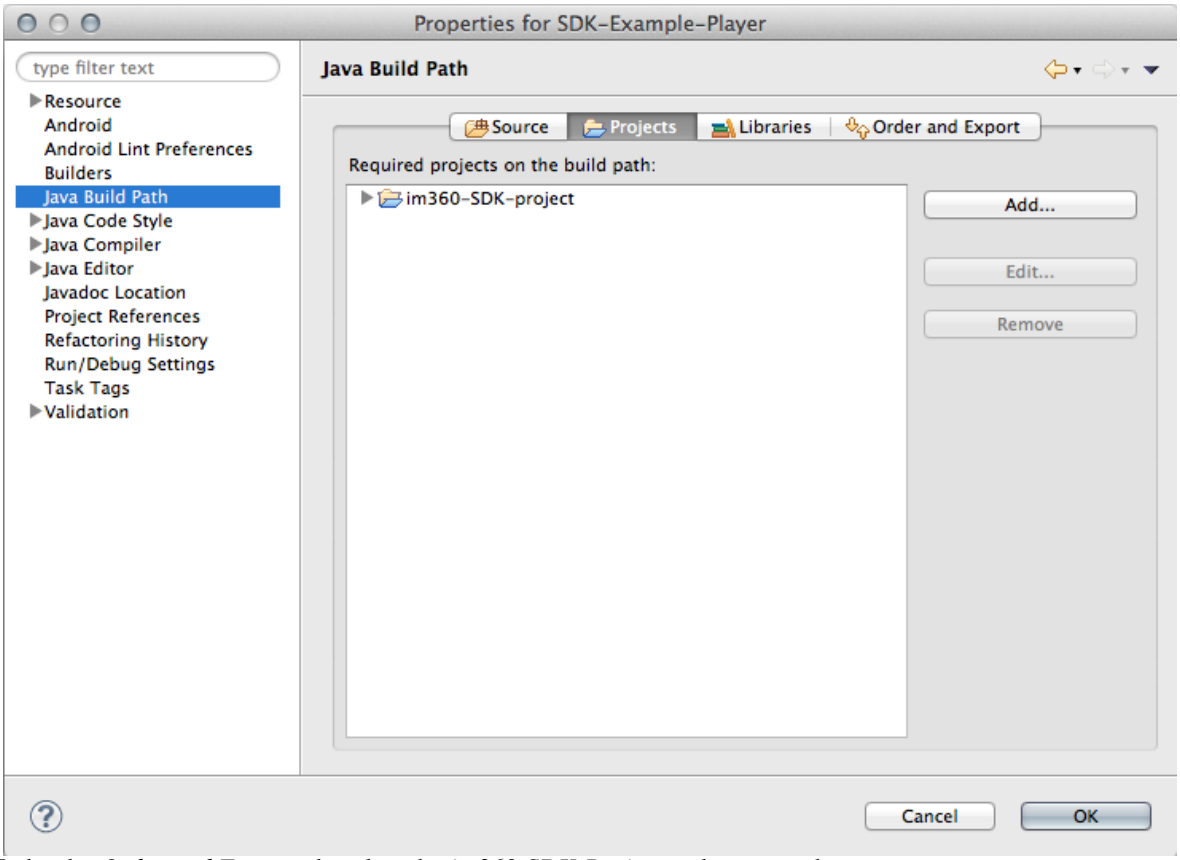
4. Once you click finish this will import the project into your workspace. Next we will import a example project.
5. Choose *File->New->Other* and choose "Android Project from Existing Code" and click "Next" like we did before.
6. Now import the *SDK-Example-Player* for the *Root Directory* and select *Finish*
7. Make sure the *im360-SDK-Project* project is set as a library. Right click on the project and choose *Properties* and Select *Android* in the left column. Make sure the *Is Library* checkbox is checked.



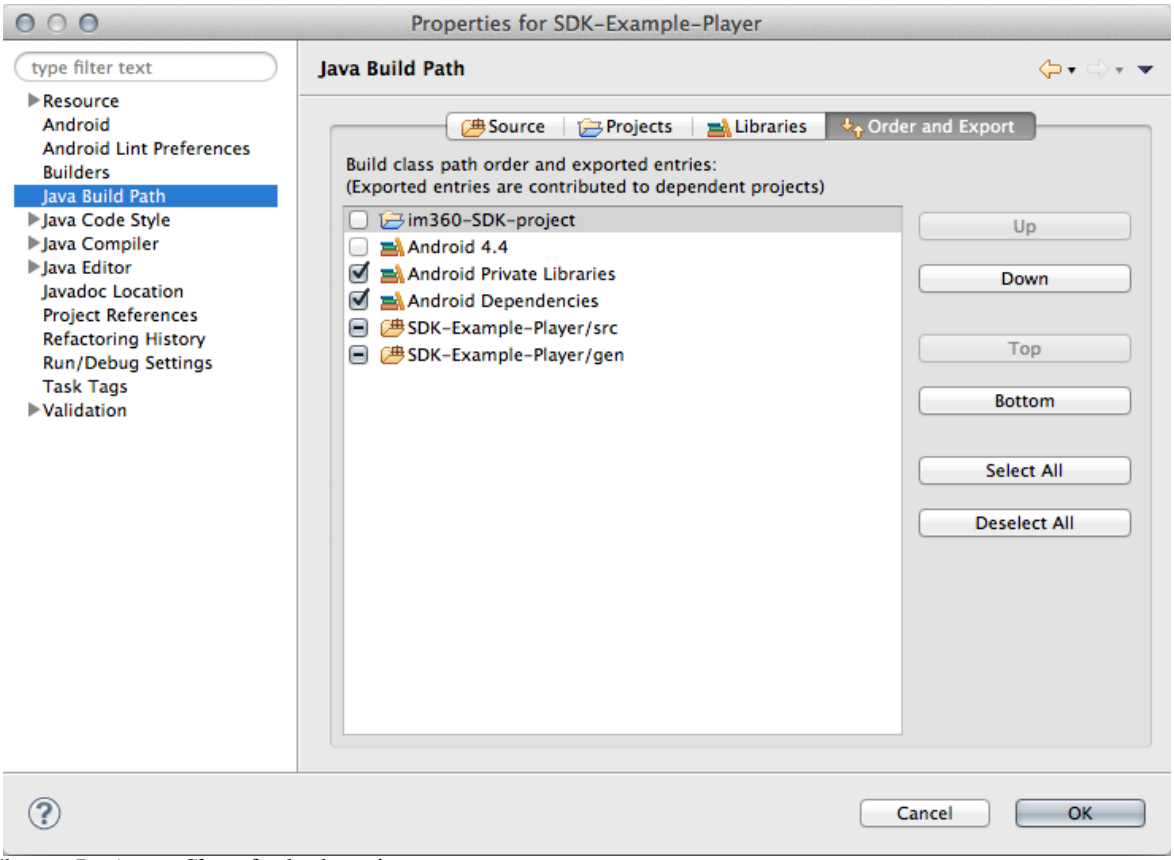
8. Set the *SDK-Example-Player* to reference the *im360-SDK-Project*. Right click on *SDK-Example-Player* and choose *Properties*. Under *Android->Library* select *Add* and choose the *im360-SDK-project*.



9. Select *Java Build Path* and under the *Projects* tab choose *Add* and select the *im360-SDK-Project*.



10. Under the *Order and Export* tab, select the *im360-SDK-Project* and move to the top.



11. Choose *Project->Clean* for both projects.
12. Choose *Project->Build* for *SDK-Example-Player*.
13. Run the *SDK-Example-Player* project.

Creating a simple Player View & playing a video

The im360 SDK project includes a built in player view and and playbar. To use it you will need to declare the *PlayerActivity* in your Manifest.

```
        <activity
                    android:name="com.im360.player.PlayerActivity"
                    android:configChanges="orientation|screenSize" > </activity>
```

Import the following packages.

```
        import com.im360.player.PlayerActivity;
        import com.im360.util.LibraryUtil;
```

In the main android activity, initialize the im360 library by calling.

```
        LibraryUtil.initData(ctx);
```

Set the Default Orientation for the given window. This lets the im360 sdk know how the gyro is orientated. Some Android devices are different then others.

```
        Config.saveDefaultOrientation(getWindow());
```

To start the *PlayerActivity* and play a remote video url it's as simple as the following.

```
      Intent intentVideo = new Intent();

      intentVideo.setClass(ctx, PlayerActivity.class);

      intentVideo.putExtra("mediaFile", "http://imc-demos.s3.amazonaws.com/media/FreeFall_1280x720f15_2M-baseline.mp4");  // media url

      startActivity(intentVideo);
```

If you would like to play a *Video Source Id* instead of a media url, use the following instead of intentVideo.putExtra("mediaFile", <url>);

```
        intentVideo.putExtra("sourceId", strSourceId);  //send the source id of video to play
```

Customizing the Player View and Playbar

PlayerActivity.java in the *im360-SDK-Project* contains the default player view. If you would like to create your own, using this class as a reference or modifying it would be a quick way of doing so. Below describes the methods in *PlayerActivity.java* and their purpose.

onCreate(Bundle savedInstanceState)

Called when the activity is first created. The *onCreate* method calls *getIntent().getExtras()* to check what arguments have been passed into the Activity. It then builds the view accordingly.

onPlayerInitialized()

onPlayerInitialized is called after the player view is initialized. This is where the player is loaded with the media file or source id. The UI timer is also started for checking on the player and updating the UI play bar with it's current state.

onStart(), onResume(), onPause(), onStop(), onDestroy()

These are the normal Android lifecycle of a activity.

OnBufferingUpdateListener()

Used to get notifications of buffering of a video.

OnInfoListener()

Get information about the video from the Android MediaPlayer.

- *MEDIA_INFO_BUFFERING_START*
- *MEDIA_INFO_BUFFERING_END*
- *MEDIA_INFO_VIDEO_RENDERING_START* (only available in Android 4.4 and above)

OnErrorListener()

Get notified if error occurred from the Android MediaPlayer.

- *MEDIA_ERROR_SERVER_DIED*
- *MEDIA_ERROR_UNKNOWN*

OnCompletionListener()

Called when the Android MediaPlayer is done playing.

OnSeekBarChangeListener()

Handles events related to the seek bar.

onTouch(View v, MotionEvent event)

OnTouch event handler for dealing with the tapping of the screen and showing/hiding the playbar.

Permissions

The *AndroidManifest.xml* must contain the following permissions

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

The *SDK-Example-Player* is good example of the requirements needed to create your own project using the im360 sdk.

Article Sources and Contributors

Im360 SDK Android 4.0 and newer *Source*: https://50.112.113.82/wiki/index.php?oldid=992 *Contributors*: Cjones

Image Sources, Licenses and Contributors

File:android-project-from-existing-code.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-project-from-existing-code.png *License*: unknown *Contributors*: Cjones
File:android-import-projects.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-import-projects.png *License*: unknown *Contributors*: Cjones
File:android-project-properties-option.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-project-properties-option.png *License*: unknown *Contributors*: Cjones
File:android-is-library.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-is-library.png *License*: unknown *Contributors*: Cjones
File:android-reference-sdk-project.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-reference-sdk-project.png *License*: unknown *Contributors*: Cjones
File:android-require-sdk-project.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-require-sdk-project.png *License*: unknown *Contributors*: Cjones
File:android-order-sdk-project.png *Source*: https://50.112.113.82/wiki/index.php?title=File:Android-order-sdk-project.png *License*: unknown *Contributors*: Cjones