Compiling for Android with Eclipse

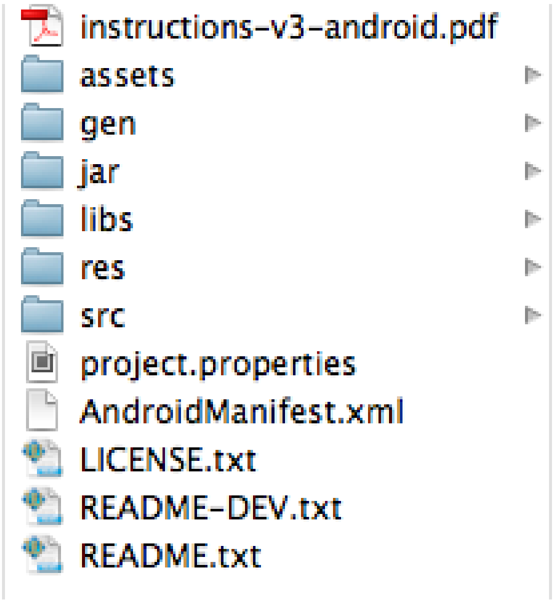
This step-by-step document assumes you have Eclipse installed on your computer along with the latest version of the Android SDK. Any version of Eclipse should work, we use “Galileo” in our workstations. You will be compiling the project source code using the Android 4.0 libraries. You will be adding a reference to the Google Play Service library in your project properties.

YOUR PROJECT WILL SHOW ERRORS UNTIL YOU COMPLETE STEP 3

1

Unzip the application's Project Files

Unzip the archive you downloaded to a safe place on your computer such as “My Documents.” Explore this location using Mac's Finder application or Windows Explorer. The directory you downloaded contains 6 sub-directories and several files.



instructions-v3-android.pdf
assets
gen
jar
libs
res
src
project.properties
AndroidManifest.xml
LICENSE.txt
README-DEV.txt
README.txt

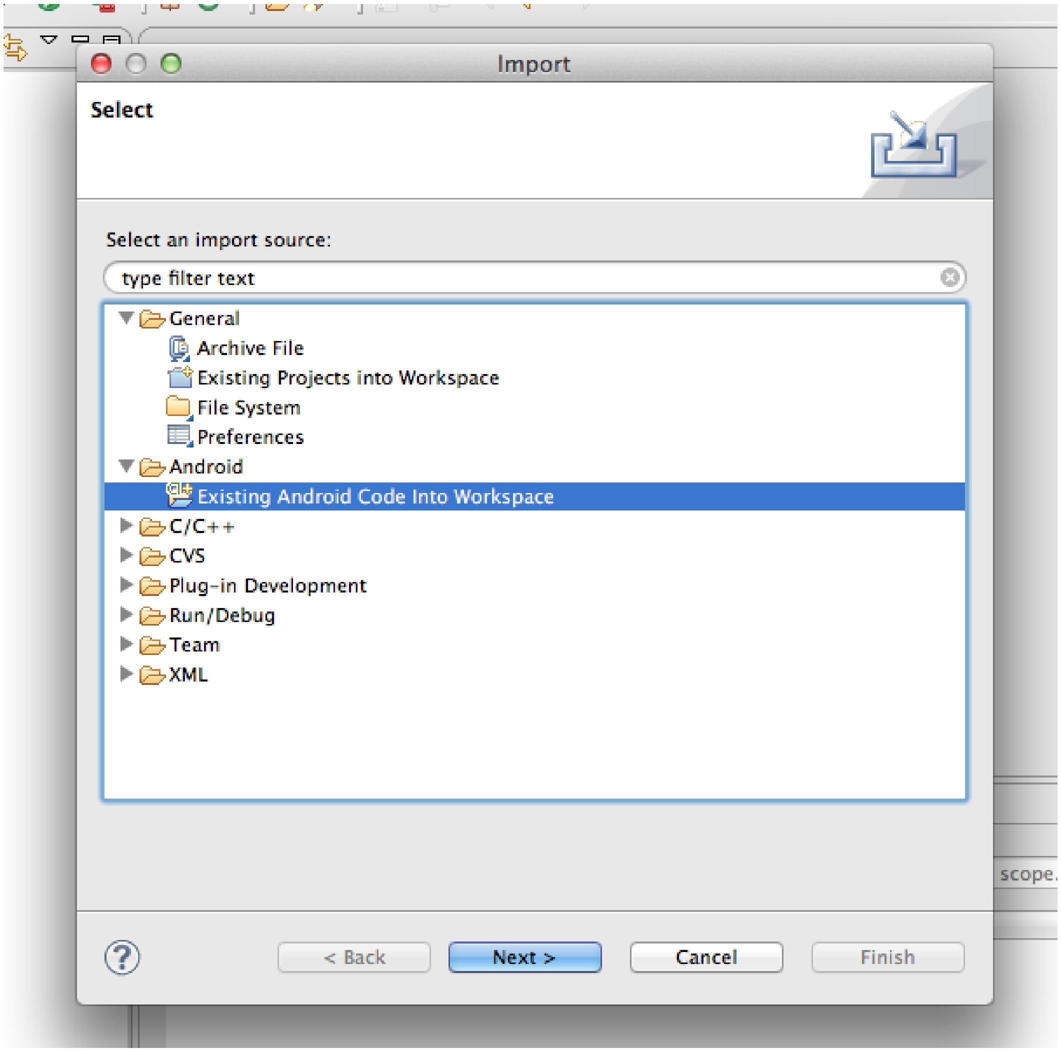
Contents of the .zip archive included in the download

2

Import your Android application into Eclipse

Launch the Eclipse development environment. When Eclipse loads you could see any number of views or panels. Eclipse refers to these views as Perspectives. Customizing your Perspective in the Eclipse IDE (Integrated Development Environment) is common. When Eclipse is finished launching, use the menu toolbar and select:

File > Import > Android > Existing Android Code Into Workspace

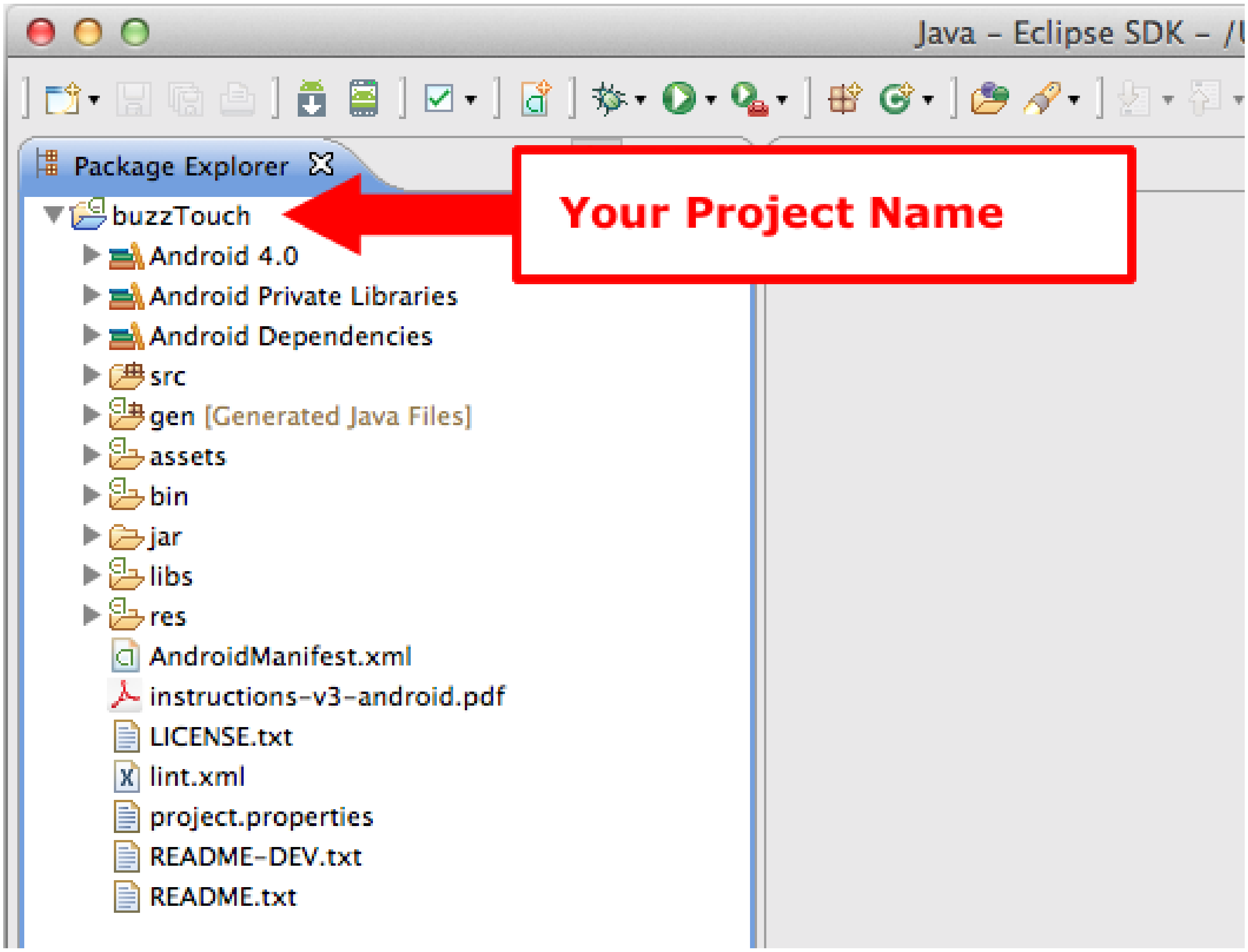


Note: If you did not install the Android SDK in Eclipse you will not see the Android > Existing Android Code into Workspace option. This means you will need to finish installing the Android SDK components before continuing.

When you choose Import > Existing Android Code into Workspace a dialogue (pop-up-window) will open. This window needs to know what project to import. You are importing the project you downloaded from your control panel.

Use the Browse button to find the folder you downloaded and unzipped. You are NOT selecting the .zip archive you downloaded, you ARE selecting the unzipped version of this download. Select the folder then click Finish.

When you're done importing the project you will see all of the project files in the Project Explorer in Eclipse. If you do not see the Project Explorer you may need to adjust Eclipse to make sure the Project Explorer panel is visible.



- buzzTouch
  - Android 4.0
  - Android Private Libraries
  - Android Dependencies
  - src
  - gen [Generated Java Files]
  - assets
  - bin
  - jar
  - libs
  - res
  - AndroidManifest.xml
  - instructions-v3-android.pdf
  - LICENSE.txt
  - lint.xml
  - project.properties
  - README-DEV.txt
  - README.txt

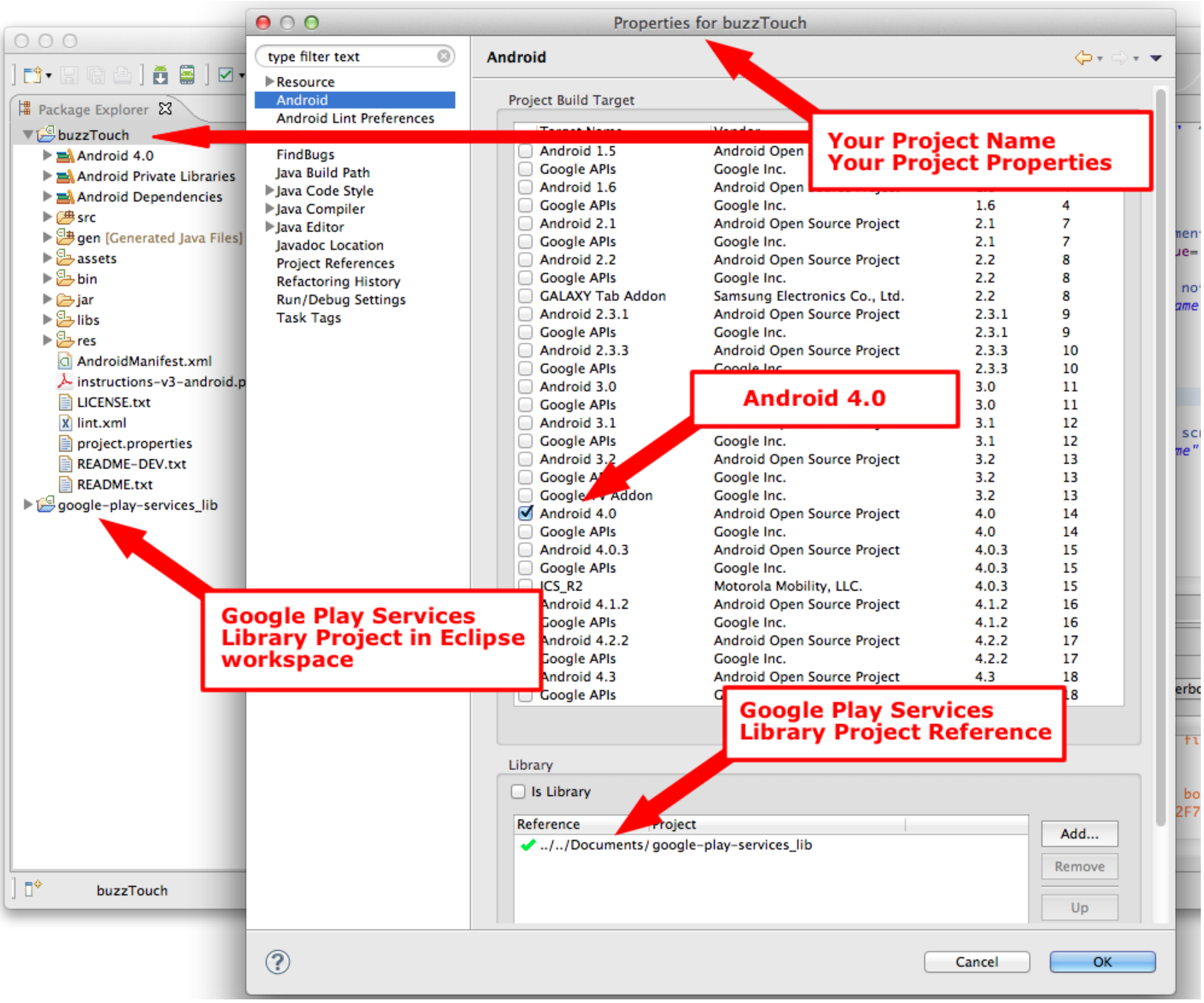
Your Project Name

Setup Google Play Services Project in Eclipse
Your Android Project relies on the Google Play Services libraries. This means you'll need to import the Google Play Services library project into your Eclipse environment before continuing. Step-by-step instructions here:

http://developer.android.com/google/play-services/setup.html

Reference the Library Project in your Android Project

Android Project Properties should look like this when you're done.



Project Build Target

	Target Name	Vendor		
□	Android 1.5	Android Open		
□	Google APIs	Google Inc.		
□	Android 1.6	Android Open		
□	Google APIs	Google Inc.	1.6	4
□	Android 2.1	Android Open Source Project	2.1	7
□	Google APIs	Google Inc.	2.1	7
□	Android 2.2	Android Open Source Project	2.2	8
□	Google APIs	Google Inc.	2.2	8
□	GALAXY Tab Addon	Samsung Electronics Co., Ltd.	2.2	8
□	Android 2.3.1	Android Open Source Project	2.3.1	9
□	Google APIs	Google Inc.	2.3.1	9
□	Android 2.3.3	Android Open Source Project	2.3.3	10
□	Google APIs	Google Inc.	2.3.3	10
□	Android 3.0		3.0	11
□	Google APIs		3.0	11
□	Android 3.1		3.1	12
□	Google APIs	Google Inc.	3.1	12
□	Android 3.2	Android Open Source Project	3.2	13
□	Google A	Google Inc.	3.2	13
□	Google Addon	Google Inc.	3.2	13
☑	Android 4.0	Android Open Source Project	4.0	14
□	Google APIs	Google Inc.	4.0	14
□	Android 4.0.3	Android Open Source Project	4.0.3	15
□	Google APIs	Google Inc.	4.0.3	15
□	ICS_R2	Motorola Mobility, LLC.	4.0.3	15
□	Android 4.1.2	Android Open Source Project	4.1.2	16
□	Google APIs	Google Inc.	4.1.2	16
□	Android 4.2.2	Android Open Source Project	4.2.2	17
□	Google APIs	Google Inc.	4.2.2	17
□	Android 4.3	Android Open Source Project	4.3	18
□	Google APIs	G		8

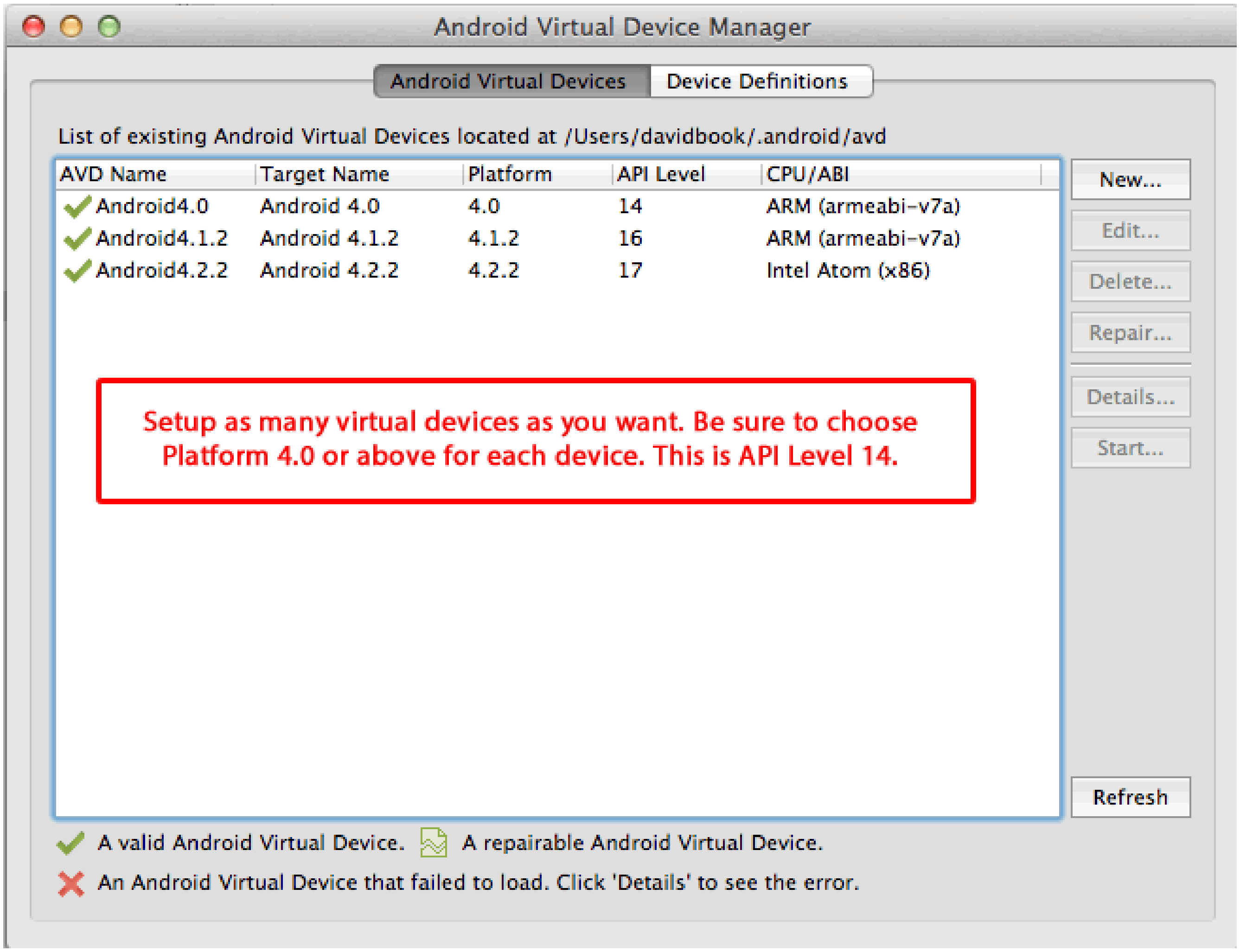
Library

□ Is Library

Reference	Project
✓ ../../Documents/ google-play-services_lib	

Setting up Virtual Devices (simulators)

Before the project can be compiled and launched you'll need to determine what device to run it on. You can run it on an actual device or an Android Virtual Device (simulator). In either case the the device must be running Android 4.0 or higher. An actual device works best but virtual devices work well too. There are some limitations on how Android Virtual Devices work but generally they do OK for testing. Use the Android Virtual Devices window to create Virtual Devices.



List of existing Android Virtual Devices located at /Users/davidbook/.android/avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ Android4.0	Android 4.0	4.0	14	ARM (armeabi-v7a)
✓ Android4.1.2	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
✓ Android4.2.2	Android 4.2.2	4.2.2	17	Intel Atom (x86)

Setup as many virtual devices as you want. Be sure to choose Platform 4.0 or above for each device. This is API Level 14.

✓ A valid Android Virtual Device. A repairable Android Virtual Device.

✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.

If you're running on an actual device you will not need to setup any virtual devices in the Android SDK and AVD Manager but make sure the actual device is connected to the computer with a USB cable. And, also, make sure the device is setup to “allow debugging” in it's settings panel.

5 Run As Android Project

With your device connected, or your virtual device running, select the projects name in the Package Explorer (single click). Then, choose Run As > Android Project from the main menu.

The Run As > Android Project menu option is tough to find sometimes. You may have a shortcut button in your toolbar, or you may have to use Eclipse's menu options. Either way you need to make sure a device is connected (or a virtual device is running) and that the project name is selected in the Package Explorer before you click Run as Android Project.

Errors:

It's possible that Eclipse will report lots and lots of errors and / or warnings when you try to Run as an Android project. There are countless reasons for this but it almost always boils down to one of these...

1) Your machine is not setup to compile Android Projects. You may think it is but hundreds of errors is a guarantee that it's not. Make sure you have the appropriate Android libraries installed in Eclipse. Use the Android SDK Manager to find and install the necessary libraries.

2) The project you setup is not compiling with the appropriate Android Libraries. This is very common. The project expects you to compile using Android 4.0 (API Level 14).

3) Your project does not have a reference to the Google Play Services library project. This reference is necessary because your project relies on features only found in that library (such as Google Maps). More info here: http://developer.android.com/google/play-services/setup.html

4) Plugins included in your project contain errors. Open source software is wonderful but errors and warnings can and do happen when lots of people are developing on a platform. If you find a warning or error in a plugin, jump on the forum and talk about it – your effort helps everyone in the Buzztouch Community.

Images and Other Assets:

Some plugins provide all the necessary images and files (audio, video, documents) required to make the plugin work, others do not. In some cases, you'll need to provide your own. If you need to provide images or other files necessary for a plugin, you need to add them to the appropriate project folder AFTER downloading your source-code from the online control panel.

If you need to add files to your project, here's where they go:

/assets/BT_Docs
Put .HTML, .PDF, .DOC, and other document based assets in this folder.
/assets/BT_Video
Put video files in this folder.
/assets/BT_Audio
Put audio files in this folder.

Video and Audio format is Android is a confusing point for lots of people. In most cases, Video is H.264 compressed and Audio is .mp3. There are lots of other supported audio / video formats. If you're unsure what format to use, consult the plugins readme.txt file or contact the plugin author.

Android Media Formats Reference:
http://developer.android.com/guide/appendix/media-formats.html