

5 reasons to choose **Streammezzo** SDK over **Android** SDK?



The purpose of this document is to give an overview of issues frequently encountered by developers when deploying an application on multiple Android phones and how these issues can be solved by Streamazzo framework.

#1 Manage multiple screen sizes and density

Android framework divides the range of actual supported screen sizes and resolutions into a set of three generalized sizes: large, normal, and small, and a set of three generalized densities: high (hdpi), medium (mdpi), and low (ldpi)

Small screen

- QVGA (240x320), 2.6"-3.0" diagonal

Normal screen

- WQVGA (240x400), 3.2"-3.5" diagonal
- FWQVGA (240x432), 3.5"-3.8" diagonal
- HVGA (320x480), 3.0"-3.5" diagonal
- WVGA (480x800), 3.3"-4.0" diagonal
- FWVGA (480x854), 3.5"-4.0" diagonal

Large screen

- WVGA (480x800), 4.8"-5.5" diagonal
- FWVGA (480x854), 5.0"-5.8" diagonal

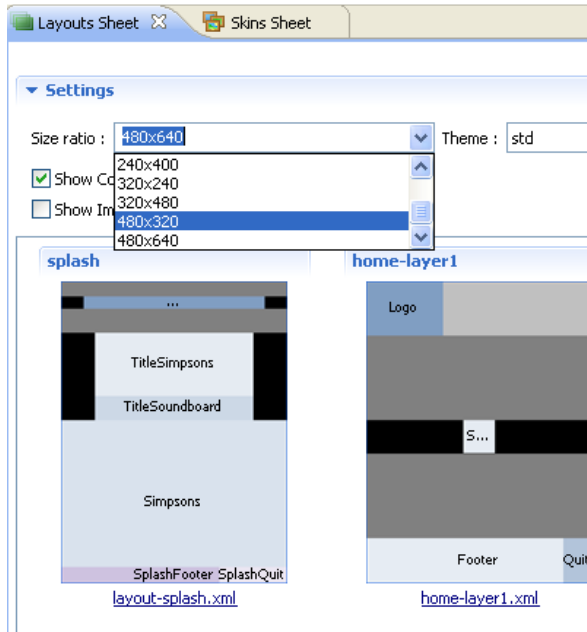
X

density (120), *ldpi* Medium density (160), *mdpi* High density (240), *hdpi*

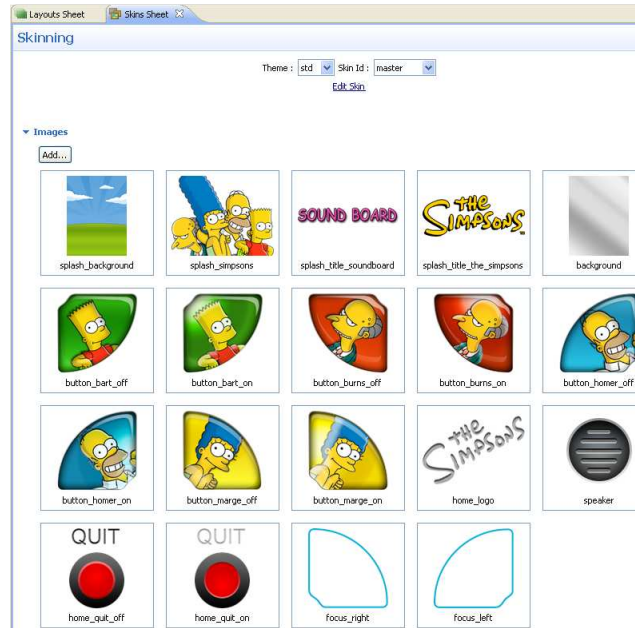


Streamazzo framework allows simpler adaptation to a broader range of screen sizes, securing application portability on future devices whether these devices are phones, tablet PC or any other connected devices.

	Android SDK	Streamazzo SDK
Designer will create a single set of image assets for the UI of the application	No, image assets should be created for all screen sizes and are bundled with the application	Yes, a single high definition image can be used and downscaled to the corresponding screen size at compilation time or server-side at runtime
Show layout in WYSIWYG	No	Yes, using the Layout Sheet
Ability to support any screen size or aspect ratio e.g. tablet device running on Android	No, pre defined screen sizes only	Yes, layout sheet allows to adapt the UI to any screen size
Image assets repository can be used by a designer	Not, image items are listed in the RES/drawable directory	Skin Sheet allows fast browsing and editing
Test application on multiple screen sizes	Specify screen size when launching the emulator android create avd ... --skin WVGA800	Change screen size at any time while emulating



Layout Sheet



Skin Sheet

#2 Profile applications according to phone capabilities

Android may run on devices ranging from low end phones to high end smart phones. Devices capabilities in term of performance, memory or features supported may vary from one phone to another.



This fragmentation misleads the developer to multiple source code or a least common denominator approach not taking advantage of all the phone capabilities. Streamezzo development framework allows profiling the application behavior while keeping a single source code to maintain.

Example 1 “Profiling your application based on navigation type”: Let’s assume that you develop an application that could use a numeric keypad in addition to touchscreen navigation. If the phone has a keypad, you would add numeric shortcuts to buttons, otherwise buttons would only be clickable using the touchscreen.



Keypad UI with numeric shortcuts



Touchscreen UI without numeric shortcuts

How would you do	
with Android SDK	with Streamezzo SDK
<p>Create a specific property file that contains the information about keypad support for each model and brand of phone</p> <p>When a new phone is launched, the property file needs to be updated otherwise default behavior would be applied to the application</p>	<p>Information about keypad support is provided by Streamezzo Device DB and can be used to profile the application behavior at runtime.</p> <pre> <!-- ***** Directional pad navigation ***** --> <% if (device.hasDirectionalPad()) { RequestParameterContext ctx = new ctx.addParameter("skin", skin); ctx.addParameter("rootLayer2", ro ctx.addParameter("soundButton1", ctx.addParameter("soundButton2", ctx.addParameter("soundButton3", ctx.addParameter("soundButton4", </pre>

Example 2 “Profiling your application based on phone performance”: Let’s assume that you develop a music application that displays a list of album covers. If the phone CPU has enough power you want display it as a cover flow with reflection effect otherwise you would choose a simple 2D carousel presentation in order to preserve power for audio playback.

How would you do	
with Android SDK	with Streamezzo SDK
<p>Create a specific property file that contains the information slow device/fast device for each brand and model of phone.</p> <p>Default behavior would be “slow device”, hence new phones not listed in the property file would not display the cover flow animation.</p>	<p>Information about phone performance is held by Streamezzo Device DB and can be used to profile the application behavior.</p> <pre> if(device.performanceLevel() == LOW) { ... } </pre> <p>New phones would instantly get the right behavior</p>

The following device information is available today: brand, model, phone OS and version, screen width and height, touchscreen, keypad, joy pad. Additional device information will be added in the next releases of the framework including: performance level, GPS support, camera support, camera position, accelerometer support, orientation sensor etc.


#3 Defragment inconsistent implementations and multiple OS versions

Some features and sensors may be implemented inconsistently on Android devices from different handset manufacturers. This fragmentation causes nightmares to application developers who have to look for patches and workarounds posted by the community or the handset manufacturers themselves.

Example 1 Camera implementation “Samsung doesn't implement the camera in the correct way-- they try to make it cooler, and it no longer works the same as other Android phones. We don't even work on some phones.” Big In Japan developer of **ShopSavvy application**, one of the top downloaded app on Android Market

Comment [8](#) by sro...@gmail.com, Feb 10, 2010

I'm thinking of a patch like this to actually parse the preview-size-values param and/or use the camera's default preview size in preference to screen resolution. Daniel, thoughts?

 **Camera_preview_size_change.patch**
4.6 KB [Download](#)

Owner: srowen

Comment [9](#) by [LatinSuD](#), Feb 10, 2010

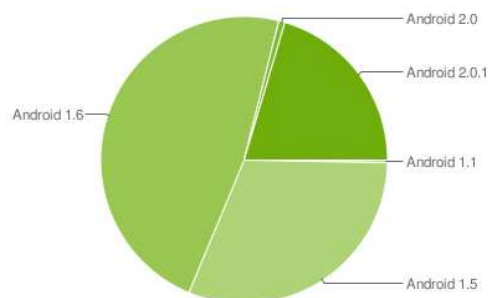
I confirm that patch works on Geeksphone (previews at 320x240) and HTC Dream (previews at 480x320).

If you were curious, these are Geeksphone current camera parameters:
antibanding=off;effect=none;jpeg-quality=85;luma-adaptation=18;picture-size=2048x1536;previe

Fragment of a discussion on code.google.com

Example 1 Android OS versions

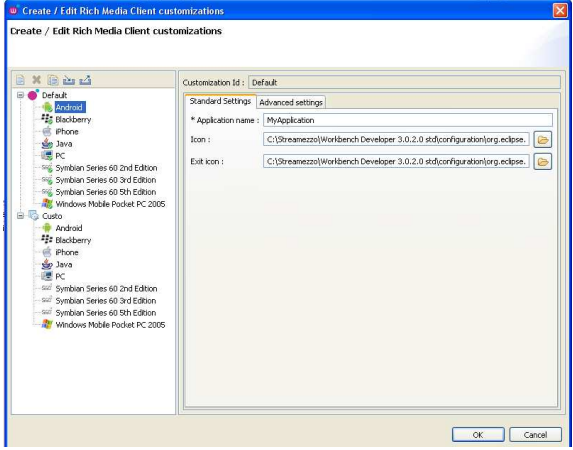
The official **Twitter application** only works on Android phones running 2.1 or newer versions of the Android operating system. That includes the Nexus One and the new HTC Droid Incredible, but doesn't include a bunch of other Android phones, including some new ones on the market



How would you do	
with Android SDK	with Streamezzo SDK
<p>Spend time looking for patches and workaround posted by the community or the handset manufacturers</p> <p>Restrict application compatibility to the latest OS versions. Developers can set a minimum API level for their applications.</p> <pre><manifest> ... <uses-sdk android:minSdkVersion="5" /> ... </manifest></pre>	<p>Streamezzo framework provides an abstraction layer ensuring that the same instruction will run consistently on all supported devices.</p> <p>e.g. camera capture command in Streamezzo <code><Action keyCodes="FIRE" url="cmd://cameraCapture"></code></p> <p>Since this abstraction layer is implemented using Android NDK, applications are not impacted by evolutions of the Dalvik APIs</p>

#4 Publish applications easily

Steps before publishing an Android application are numerous and error prone. Streamezzo framework allows developers to create their .apk package in just few clicks:

How would you do	
with Android SDK	with Streamezzo SDK
<p>Developer should edit one manifest file and run four different tools in order to get the .apk package</p> <p>Specify an icon and label: Define the attributes <code>android:icon</code> and <code>android:label</code> in the <code><application></code> element of the manifest</p> <p>Remove the <code>android:debuggable="true"</code> attribute from the <code><application></code> element of the manifest.</p> <p>Define a version number for your application, specifying an appropriate value for both the <code>android:versionCode</code> and <code>android:versionName</code> attributes of the <code><manifest></code></p> <p>Obtain a suitable cryptographic key To generate a self-signed key with Keytool</p> <p>Compile your application using ADT plugin</p> <p>Sign the application using Keytool and Jarsigner.</p> <p>Use the zipalign tool to optimize the final APK package.</p>	<p>All the steps required to create the .apk file are available in a single visual tool allowing 1-click deployment once the application information has been filled in.</p> 

#5 Manage audio and video playback

Video playback capabilities differ from one handset to another. Streamezzo framework comes with its own video player providing a consistent behavior across all supported handsets.

END OF DOCUMENT