

Mobile and Wireless Systems Programming

Introduction to Android



- Android is a software stack for mobile devices that includes :
 - an operating system
 - middleware
 - key applications
- Open source project based on Linux kernel 2.6
- Open Handset Alliance (Google, HTC, Samsung...)
- Official website : <http://www.android.com>

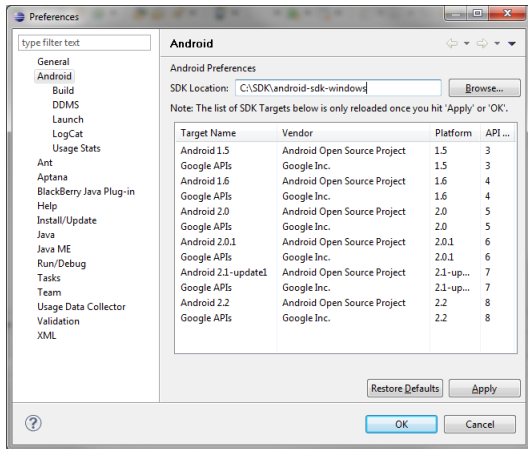
- 1 binary for all devices
- Supports multi-resolutions
- Java 1.5 (Collections, annotations,)
- Rich API for developers
- ...



- **Application** framework enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source WebKit engine
- **Optimized graphics** powered by a custom 2D graphics library ; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, and WiFi** (hardware dependent)
- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

- ADT : Android Development Tools (Eclipse Plugin)
 - Project Wizard
 - UI builder
 - Resource manager
 - Debug, compile, package, sign
 - Views + Perspectives
 - ...
- Android SDK
 - APIs
 - Emulators
 - Tools
 - Samples (API Demos)
 - ...



- **add-ons/**
- **docs/**
- **platforms/** : set of Android platform versions
 - **platform/** : for example "android-1.6"
 - **data/** : Storage area for default fonts and resource definitions.
 - **images/** : Images (disk, Android system,...) for the emulator sessions.
 - **skins/** : emulator skins
 - **templates/**
 - **tools/**
 - **android.jar** : Android library used when compiling applications
- **samples/** : Sample code and apps
- **tools/** : emulator, the android tool, adb, ddms...
- **SDK Readme.txt**
- **SDK Setup.exe** : Android SDK and AVD Manager tool

emulator.exe

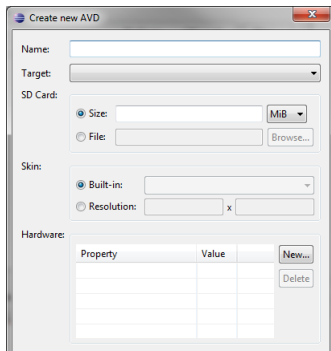
- Android emulator
- Based on QEMU
- Actual Android run-time environment
- TouchScreen, Buttons : Home, Back...

Android Emulator



AVD

- Virtual device configuration
- Hardware options, emulator skin...
- i.e. test an application with different screen size



Hierarchy Viewer

- Debug and optimize UI
- Visual representation of layout's hierarchy

The screenshot shows the Hierarchy Viewer application. The main window displays a tree view of the layout hierarchy. The root node is `PhoneWindowDecorView` (android.support.design.widget.AppBarLayout\$DecorView). Below it is `AppBarLayout$DecorView`, then `AppBarLayout$DecorView`. The tree branches into several `TextView` nodes and `CollapsingToolbarLayout` nodes. The right-hand panel shows the properties of the selected node, including `absolute_x`, `absolute_y`, `getWidth()`, `getHeight()`, `isEnabled()`, `isFocusable()`, `isFocusableInTouchMode()`, `layout_height`, `layout_width`, `layout_weight`, and `layout_gravity`. The bottom right corner shows a preview of the UI with a red box highlighting a specific area.

Property	Value
<code>absolute_x</code>	61
<code>absolute_y</code>	227
<code>getBaseline()</code>	28
<code>getHeight()</code>	48
<code>getTag()</code>	null
<code>getVisibility()</code>	VISIBLE
<code>getWidth()</code>	188
<code>hasFocus()</code>	true
<code>isClickable()</code>	true
<code>isDrawingCacheEnabled()</code>	false
<code>isEnabled()</code>	true
<code>isFocusable()</code>	true
<code>isFocusableInTouchMode()</code>	true
<code>isSelected()</code>	false
<code>layout_bottomMargin</code>	0
<code>layout_gravity</code>	NONE
<code>layout_height</code>	WRAP_CONTENT
<code>layout_leftMargin</code>	0
<code>layout_rightMargin</code>	0
<code>layout_topMargin</code>	0
<code>layout_weight</code>	1.0

layoutopt

- command-line tool t
- optimize layouts
- Example : `> layoutopt res/layout-land`

DDMS

- A debugging tool
- provides port-forwarding services
- screen capture on the device
- thread and heap information on the device
- logcat, process, and radio state information,
- incoming call and SMS spoofing, location data spoofing
- ...

ADB

- Un/Install .apk files on an emulator or device
- Access the emulator or device from a command line
- Link a debugger to application running on an emulator or device
- ...

- **Traceview** : produces graphical analysis views of trace log data
- **sqlite3** : access the SQLite data files created and used by Android applications
- **Draw 9-patch** : create a NinePatch graphic using a WYSIWYG editor
- **aapt** : create .apk files
- **aidl** : generate code for an interprocess interface
- **mksdcard** : create a disk image to simulate the presence of an external storage card
- **dx** : rewrites .class bytecode into Android bytecode
- **UI/Application Exerciser Monkey** : stress-test applications
- **android** : a script to manage AVDs
- **zipalign** : .apk optimization tool

- All applications have to be signed
- Android Tools → Export Signed Application package. . .
- Self-signed certificates
- keytool and jarsigner
- Code and Application versionning
 - android :versionCode
 - android :versionName
- Android compatible platform (1.5, 1.6, 2.0, 2.1, 2.2. . .)
- Register and sell your app on the Android Market

- Develop and test for different resolutions (use AVD)
- Userfriendly, well documented and huge community
- **One link** : <http://developer.android.com>