



Technology Consulting Company
Research, Development &
Global Standard

Introduction to Android

Takanari Hayama

taki@igel.co.jp

STARTING WITH ANDROID

Android上での開発

■ SDK – Software Development Kit

- アプリケーション開発用
- Java言語ベースのアプリケーション開発
- <http://developer.android.com/>

■ PDK – Platform Development Kit

- プラットフォーム開発用
- 新規ハードウェアのサポートやJava言語だけでは開発できない新規サービスの開発
- <http://source.android.com/>

■ プラットフォーム開発に便利なツールもSDKに含まれているので、SDKは必ず入手すること

エミュレータの初期化

- C:\¥android-sdkにSDKをインストールし、cygwin環境を使っていることを想定

```
$ cd /cygdrive/c/android-sdk/tools
```

```
$ ./android.bat list target
```

```
Available Android targets:
```

```
id: 1
```

```
  Name: Android 1.1
```

```
  Type: Platform
```

```
  API level: 2
```

```
  Skins: HVGA (default), HVGA-L, HVGA-P, QVGA-L, QVGA-P
```

```
id: 2
```

```
  Name: Android 1.5
```

```
  Type: Platform
```

```
  API level: 3
```

```
  Skins: HVGA (default), HVGA-L, HVGA-P, QVGA-L, QVGA-P
```

```
id: 3
```

```
  Name: Google APIs
```

```
  Type: Add-On
```

```
  Vendor: Google Inc.
```

```
  Description: Android + Google APIs
```

```
  Based on Android 1.5 (API level 3)
```

```
  Libraries:
```

```
    * com.google.android.maps (maps.jar)
```

```
      API for Google Maps
```

```
  Skins: QVGA-P, HVGA-L, HVGA (default), QVGA-L, HVGA-P
```

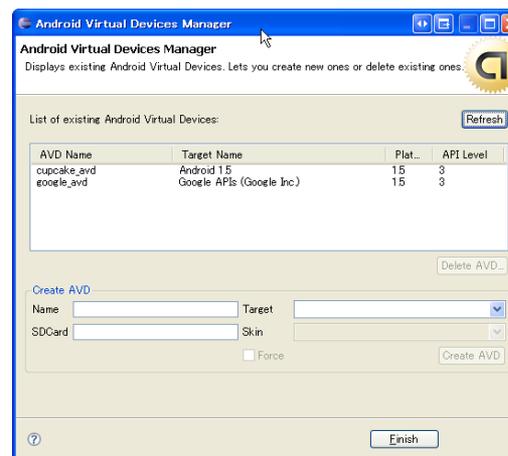
エミュレータの初期化 (contd.)

■ Google API 付きで仮想デバイスを作成

```
$ ./android.bat create avd --target 3 --name google_avd
Created AVD 'google_avd' based on Google APIs (Google Inc.)
$ ./emulator.exe -avd google_avd &
エミュレータが起動する。
$ ./adb -e shell
# cat /proc/version
cat /proc/version
Linux version 2.6.27-00110-g132305e (mikechan@cheetara.corp.google.com) (gcc
version 4.2.1) #6 Mon Feb 2 12:47:38 PST 2009
```

■ Eclipseに最新版のAndroidプラグインを入れるとGUIでエミュレータ環境の初期化が可能

– 「Window」→「Android AVD Manager」もしくは「ケータイ・アイコン」



便利なコマンド adb

- android-sdk/tools以下にエミュレータやデバッグに必要なツールあり
- デバイスへのログイン
 - \$ adb -e shell (エミュレータへのログイン)
 - \$ adb -d shell (デバイスへのログイン)
- デバイスのログ表示
 - \$ adb -d logcat
- ファイルの読み書き
 - \$ adb -d push <local> <remote>
 - \$ adb -d pull <remote> <local>

Android Dev Phone 1のファーム書き換え

■ 書き換え手順はシンプル

- <http://www.htc.com/www/support/android/adp.html>

■ 基本的なステップ

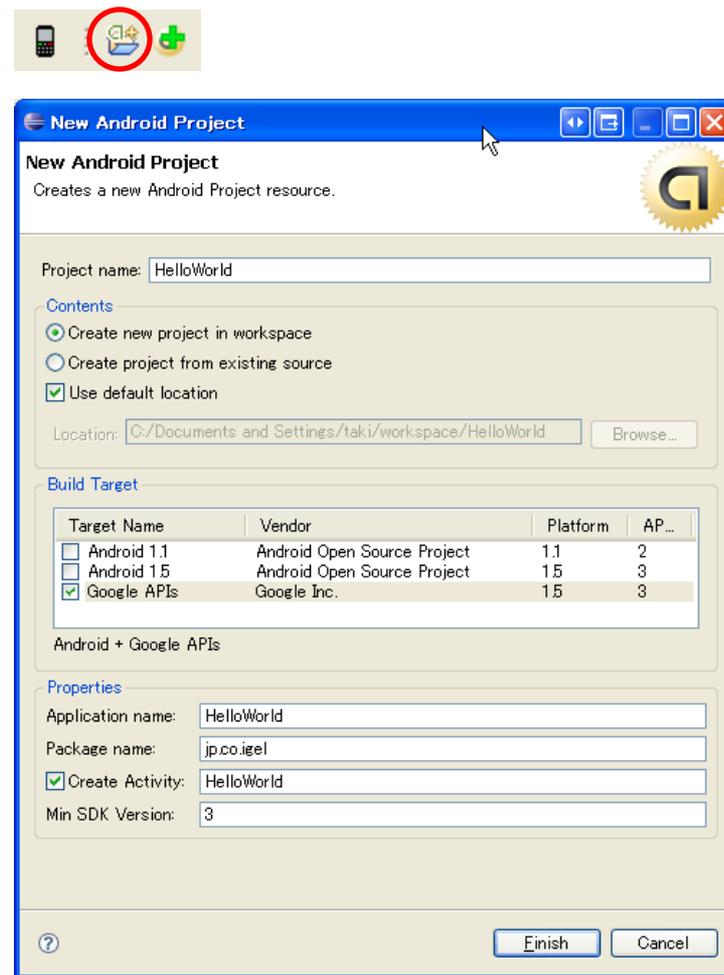
1. Radio ImageとRecovery Imageをダウンロード
2. Radio Imageをadb pushで、/sdcard/update.zipにコピー
3. HOMEキーを押しながリブート(電源投入)し、「！」が表示されたら、Alt+l (ログ表示)、Alt+s (ファーム書換)を入力
4. ファームがアップデートされたら、HOMEキーとBACKキーを同時押ししリブート
5. Recovery Imageで、2からのステップを実施

- ログイン・プロンプトが「\$」の場合、root権限が無いので「su」でrootになること
 - パスワードは不要
 - プロンプトが「#」に
- /system/xbin以下にデバッグに便利なツールあり
 - strace、ps、top など
- Windows環境のadbでは、Ctrl+Cをするとadb自体が終了してしまうので注意すること
 - プロンプトが「\$」になるので判りにくい。。。。

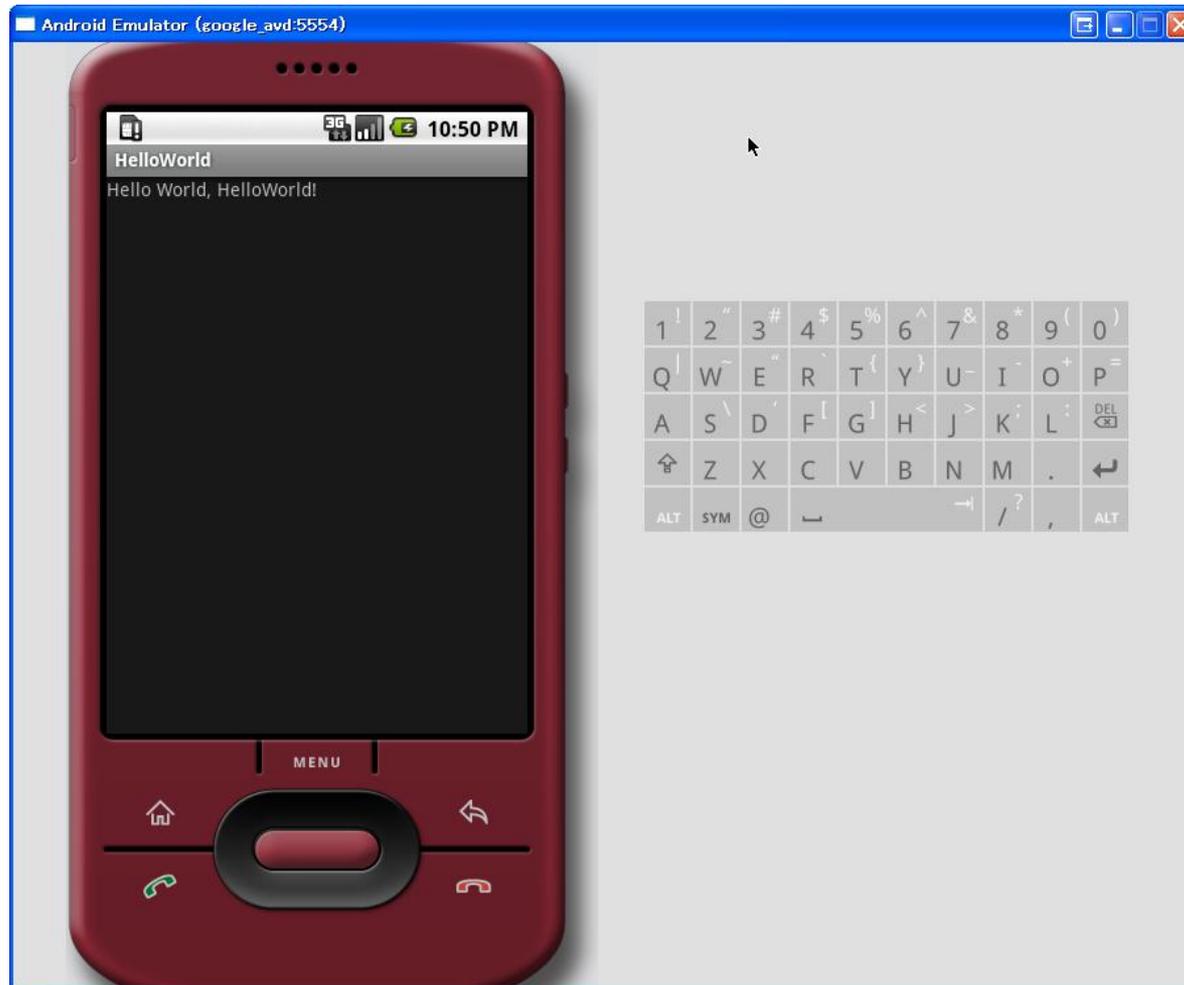
DEVELOPING APPLICATIONS

Hello World

- Android Project Wizardを利用
- とりあえず、必要な情報を入力
 - 左のダイアログを参考に
 - プロジェクトができれば、とりあえず何も考えずに実行
 - Android Applicationとして起動
 - MacOS X環境の場合は、一度、起動に失敗するので、「Run」→「Run Configuration...」で、Android ApplicationからHelloWorldなどの作成したアプリを選択し、「Common」タブの「Save as」で「Shared file」にすること
 - 権限の関係で起動に失敗



起動した Hello World



ちゃんとしたTutorial

- Androidのドキュメント類はきちんと揃っているので、以下を参考に
 - <http://developer.android.com/guide/index.html>
- もう少し真面目なHello Worldは
 - <http://developer.android.com/guide/tutorials/hello-world.html>
- 他にも色々なサンプルやチュートリアルがあるので必見

PLAYING WITH SOURCE CODE

- とりあえずは、<http://source.android.com/download> に沿ってやってみる。
 - とりあえず、Ubuntu 8.10 上では問題無く動作済み
 - repoの使い方は<http://source.android.com/download/using-repo>を参考
- ソースを最新の状態に維持
 - `repo sync`
- 自分でコードを追加したり、修正する場合
 - `repo start branchname project`
 - 例: `repo start igell kernel`
 - 後は通常通りgitコマンドを利用

HTC用のコードなど

- repo initするだけでは、入手できないプロジェクトもあり
 - プロジェクト一覧は<http://android.git.kernel.org>を参照
- .repo/local_manifest.xmlを記述することで標準以外のプロジェクトをrepoで管理機可能に
 - 標準のmanifestはgitで管理されているので変更しないこと
 - <http://android.git.kernel.org/?p=platform/manifest.git;a=summary>
- .repo/local_manifest.xmlの例

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest>
  <remove-project name="kernel/common"/>
  <!-- <project path="kernel" name="kernel/msm"
    revision="refs/heads/android-msm-htc-2.6.25"/> -->
  <project path="kernel" name="kernel/msm" revision="refs/heads/android-
    msm-2.6.29"/>
  <project path="external/alsa-lib" name="platform/external/alsa-lib"/>
  <project path="external/alsa-utils" name="platform/external/alsa-
    utils"/>
  <project path="vendor/htc/dream" name="platform/vendor/htc/dream"/>
  <project path="vendor/qcom/surf" name="platform/vendor/qcom/surf"/>
  <project path="vendor/asus/eee_701"
    name="platform/vendor/asus/eee_701"/>
  <project path="hardware/msm7k" name="platform/hardware/msm7k"/>
  <project path="hardware/alsa_sound"
    name="platform/hardware/alsa_sound"/>
</manifest>
```

■ HTC Android端末用カーネル・コード

- kernel/arch/arm/mach-msm/
 - HTC Magic (DoCoMo/Vodafone端末): board-sapphire*
 - HTC Dream (aka G1/Dev Phone 1): board-trout*

■ SDK 1.5で使われているCupcakeのカーネルは、2.6.27ベース

- 入手可能なブランチ一覧は、kernelディレクトリ以下で:
 - # `git branch -a`
- ブランチの変更は:
 - # `git checkout korg/android-msm-2.6.29`

ANDROID PORTING DEVELOPMENT KIT (PDK)

- Android移植のための開発キット
- ドキュメントは以下の手順で作成可能
 - development/pdk/READMEを必ず読むこと
 - doxygenやjavaが必要
 - % cd <repo initしたディレクトリ>
 - % . build/envsetup.sh
 - % mkdir dist
 - % mkdir logs
 - % time make -j4 pdk pdk_all dist DIST_DIR=dist 2>&1 | tee logs/`date +%y%m%d-%H%M%S`
- 成功すると out/target/common/docs/online-pdk にドキュメント
 - dist以下にはtgzファイルもあり
 - ベンダ依存のコードを持ってきていると失敗する場合がありますので、その場合は、「mv ./vendor ..」として一時的に退避させること
 - 移植に必要な情報のヒントにはなるので目を通しておくと良いが、最新じゃないので要注意
- ブラウズするにはGoogle App Engineが必要
 - <http://code.google.com/appengine/downloads.html>
 - ダウンロードしたファイルをunzipし、google_appengineディレクトリから:
 - % ./dev_appserver.py <android_rootdir>/out/target/common/docs/
 - % firefox <http://localhost:8080/>
- ヒント
 - dev_appserver.py に「-a 0.0.0.0」を追加すると有効になっている全てのIPアドレスにバインドされる。デフォルトはlocalhostのみ。
 - 全てのドキュメントが見られない(リンクが落ちている)ので、個別にファイルを開く必要があるケースもあり(電源管理系やカメラに関する情報など)
 - <http://localhost:8080/online-pdk/camera.html>
 - http://localhost:8080/online-pdk/power_management.html

ANDROID INTERNAL

- Power Management を始め、様々なシステム・サービスの親玉
 - adb -d logcat でログを出しながらプロセスの辺りをつけ
 - strace -p 62 -f で調査
 - 開いているファイルなどは地道にlsで調査
 - ls -l /proc/<pid>/fd/<fd>
- ブート順序
 - init -> zygote -> SystemServer
 - ./frameworks/base/cmds/system_server 以下にソース
 - 実際には、com/android/server/SystemServerをインスタンス化しているだけ
 - ./frameworks/base/services/java/com/android/server/SystemServer.javaがシステム・サーバの実体
 - ちなみに、zygoteはクラス・ライブラリをpreloadしていて、全てのandroid上のJavaアプリケーションは、zygoteの子プロセスとして実行
 - クラス・ライブラリのロード時間圧縮

- SystemServerから起動されるPowerManagerServiceが管理の実体
 - 基本的にはバックライトやバッテリー・レベルに応じたデバイスの簡単な制御程度？
 - 要調査
 - `./out/target/common/docs/online-pdk/power_management.html` にアーキテクチャ
 - 管理サービスは以下で実装:
 - `./frameworks/base/services/java/com/android/server/PowerManagerService.java`
- バックライトの制御は `/sys/class/leds` 以下

```
# ls /sys/class/leds
lcd-backlight
blue
green
red
spotlight
keyboard-backlight
button-backlight
```
- バックライト制御は:
 - 通常

```
# echo 102 > /sys/class/leds/lcd-backlight/brightness
```
 - ディム

```
# echo 20 > /sys/class/leds/lcd-backlight/birghtness
```
 - オフ

```
# echo 0 > /sys/class/leds/lcd-backlight/birghtness
```

- ./frameworks/base/libs/SurfaceFlinger
 - SurfaceFlinger -> PixelFlinger -> OpenGL ES
- OpenGL ESサポートが無いとソフトウェア描画に
 - 現在は、OpenGL ES 1.1のみだが2.0もサポート予定？
 - SurfaceFlinger自体はES 2.0対応しているか不明

■ PacketVideo社のOpenCOREを採用

- Apacheライセンスだから、というだけで、公式(?)にはGstreamerが性能・機能的に悪いから、ということではないとのこと
 - http://groups.google.com/group/android-platform/browse_thread/thread/e3ba69545e1d31ac/b7ffedb401fe0756
- Media Serverデーモンが起動していて、基本的にMedia ServerがIPCでリクエストを受け付け再生
 - OMA DRMなど基本的なものはサポート済み
- external/opencoreにOpenCOREのコード

■ AudioにはALSAが追加されているがステータスは不明

- 独立したプロジェクト

デフォルトでサポートされているメディア



Type	Format	Encoder	Decoder	Details	File Type(s) Supported
Audio	AAC LC/LTP		X	Mono/Stereo content in any combination of standard bit rates up to 160 kbps and sampling rates from 8 to 48kHz	3GPP (.3gp) and MPEG-4 (.mp4, .m4a). No support for raw AAC (.aac)
	HE-AACv1 (AAC+)		X		
	HE-AACv2 (enhanced AAC+)		X		
	AMR-NB	X	X	4.75 to 12.2 kbps sampled @ 8kHz	3GPP (.3gp)
	AMR-WB		X	9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz	3GPP (.3gp)
	MP3		X	Mono/Stereo 8-320Kbps constant (CBR) or variable bit-rate (VBR)	MP3 (.mp3)
	MIDI		X	MIDI Type 0 and 1. DLS Version 1 and 2. XMF and Mobile XMF. Support for ringtone formats RTTTL/RTX, OTA, and iMelody	Type 0 and 1 (.mid, .xmf, .mxmf). Also RTTTL/RTX (.rtttl, .rtx), OTA (.ota), and iMelody (.imy)
	Ogg Vorbis		X		Ogg (.ogg)
PCM/WAVE		X	8- and 16-bit linear PCM (rates up to limit of hardware)	WAVE (.wav)	
Image	JPEG	X	X	Base+progressive	JPEG (.jpg)
	GIF		X		GIF (.gif)
	PNG		X		PNG (.png)
	BMP		X		BMP (.bmp)
Video	H.263	X	X		3GPP (.3gp)
	H.264 AVC		X		3GPP (.3gp) and MPEG-4 (.mp4)
	MPEG-4 SP		X		3GPP (.3gp)

※ <http://developer.android.com/guide/appendix/media-formats.html>より

T-Mobile G1で追加サポートされているフォーマット



Type	Format	Encoder	Decoder	Details	File Type(s) Supported
Audio	WMA		X	<ul style="list-style-type: none"> •Supports WMA standard L1-L3: L1: 64 kbps - 161 kbps @ 44.1kHz •L2: <=161 kbps <=48 kHz •L3: <385 kbps <=48 kHz Mono and stereo profiles with 16-bits per sample. Decoder does not support WMA Pro, Lossless, or Speech codecs.	Windows Media Audio (.wma)
Video	WMV		X	Versions 7, 8 and 9. Simple profile only	Windows Media Video (.wmv)
	H.263	X	X		3GPP (.3gp) and MPEG-4 (.mp4)
	H.264 AVC		X	Limited to baseline profile up to 480x320, and 600 kbps average bitrate for the video stream.	3GPP (.3gp) and MPEG-4 (.mp4)

- ライセンス上の問題でWMV形式はAndroid Platform外
- G1ではH.263でMPEG-4形式を追加サポート
- G1ではH.264 AVCのサポート・プロファイルを480x320, 600kbpsに限定