

Hướng dẫn lập trình cơ bản với Android - Bài 0

List tutorial

Bài 0 - Cài đặt và sử dụng Android với Eclipse

Bài 1 - Cơ bản Android

Bài 2 - Xây dựng giao diện đơn giản

Bài 3 - ViewGroup và Custom Adapter

Bài 4 - Intent và Broadcast Receiver

Bài 5 - Service

Bài 6 - SQLite

Bài 7 - Content Provider

Hi mọi người,

Trong lúc chờ đợi anh Thông release bài 8, mình quyết định viết bài 0 này, bởi vì có khá nhiều newbie hỏi mình về các vấn đề cơ bản như sử dụng Logcat, sử dụng Device, làm sao xoay ngang màn hình hay chụp screenshot. Để tránh tình trạng phải trả lời đi trả lời lại nhiều lần, bài này sẽ hướng dẫn cặn kẽ mọi người cách sử dụng Android trong Eclipse, hi vọng giúp những người mới chập chững bước vào lập trình Android có thêm thông tin và biết cách khai thác IDE của mình hiệu quả hơn.

I. Hướng dẫn cài đặt Android với Eclipse:

Thực chất anh Giáp đã viết một bài có nội dung tương tự, nhưng do Google thay đổi phương thức tải Android SDK nên mình quyết định viết lại, gộp luôn vào topic hướng dẫn sử dụng Eclipse.

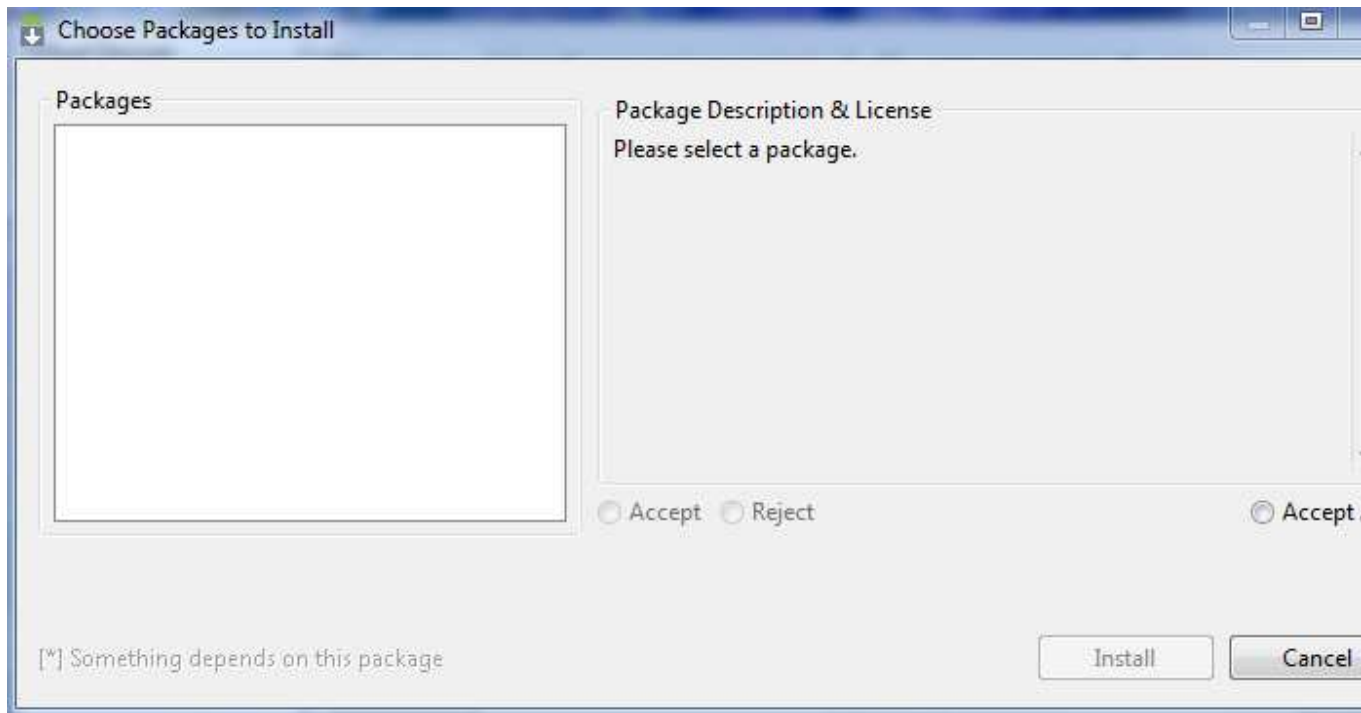
1. Download Android SDK:

Android SDK thực chất là tập hợp các công cụ và thư viện để phát triển các ứng dụng trên nền tảng hệ điều hành Android.

B1: Vào trang <http://developer.android.com/sdk/index.html> để tải Android SDK Starter. Tùy thuộc vào hệ điều hành mà bạn chọn bản Mac, Linux hay Window. Ở đây mình chọn tải bản cho Window.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r06-windows.zip	23293160 bytes	7c7fcec3c6b5c7c3df6ae654b27effb5
Mac OS X (intel)	android-sdk_r06-mac_86.zip	19108077 bytes	c92abf66a82c7a3f2b8493ebe025dd22
Linux (i386)	android-sdk_r06-linux_86.tgz	16971139 bytes	848371e4bf068dbb582b709f4e56d903

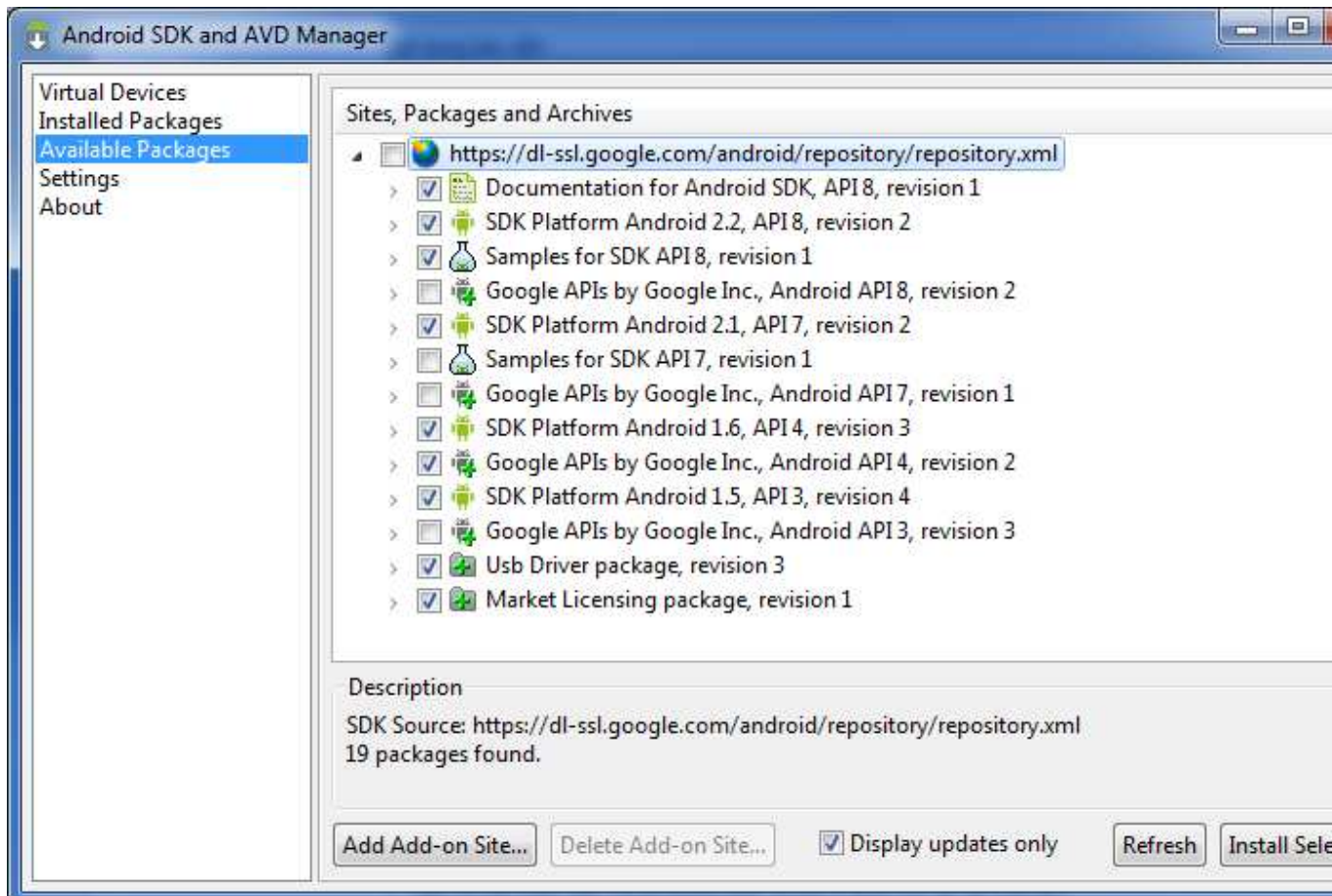
B2: Giải nén file zip bạn vừa tải về. Chạy SDK Setup.exe. Bạn có thể gặp thông báo lỗi Fetching <https://dl-sl...> Failed to fetch... Close thông báo này lại. Tiếp theo cửa sổ Choose Packages to Install xuất hiện. Nếu cửa sổ này trống rỗng -> Cancel.
-> Quay về cửa sổ Android SDK and AVD manager -> Chọn Setting, đánh dấu vào ô Force <https://...>
-> Chọn Available Packages



B3: Đánh dấu các Packages bạn muốn tải: Documents chính là phần Javadoc mô tả hoạt động của các phương thức và các lớp (phần này chắc chắn không thể thiếu rồi), Sample là các đoạn code mẫu, SDK Platform ứng với các phiên bản hệ điều hành (2.2 - API level 8, 2.1 - API level 7,...), và Google API để phát triển các phần mềm liên quan đến dịch vụ của Google (như Google Map nếu bạn muốn lập trình liên quan đến GPS).

Các bạn có thể tải hết nếu thích, còn muốn tối ưu thì có thể đánh dấu như mình (lưu ý USB drivers chỉ dành cho người sử dụng Windows và muốn phát triển ứng dụng test bằng điện thoại thật).

- > Install Selected
- > Install
- > Cửa sổ Install hiện ra
- > Ngồi chờ (>_<)



2. Tích hợp Android SDK vào Eclipse:

B1: Tải Eclipse nếu bạn chưa có. Mọi người có thể phân vân không biết tải bản nào cho phù hợp, nhưng theo ý kiến của mình thì có thể dùng 1 trong 2 bản sau: Eclipse for Java Developers, hoặc Eclipse for Java and Report Developers (mình dùng bản sau).

B2: Khởi chạy Eclipse, vào Help -> Install new softwares.

Chọn Add, gõ vào ô Name tên bạn muốn và Location gõ vào địa chỉ để tải về ADT:

HTML Code:

```
https://dl-ssl.google.com/android/eclipse/
```

hoặc

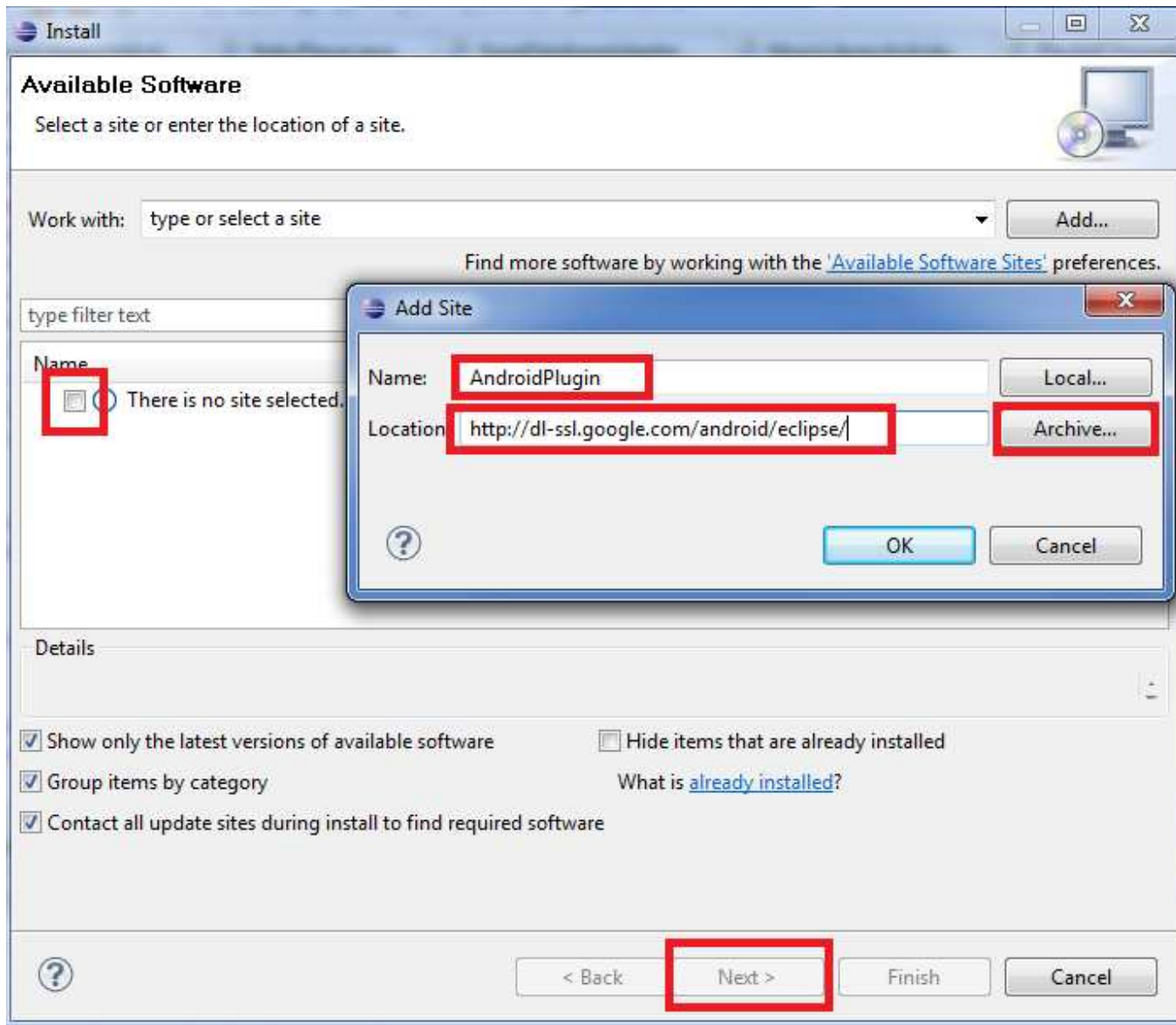
HTML Code:

```
http://dl-ssl.google.com/android/eclipse/
```

nếu https không hoạt động. Ngoài ra bạn cũng có thể tải thẳng ADT về máy theo link <http://dl.google.com/android/ADT-0.9.7.zip> (bản mới nhất 0.9.7 ứng với Android 2.2), chọn Archive và browse tới file này (lưu ý không giải nén)

-> OK

-> Check vào phần dưới ô Name (sẽ hiện ra dòng Developer Tools).



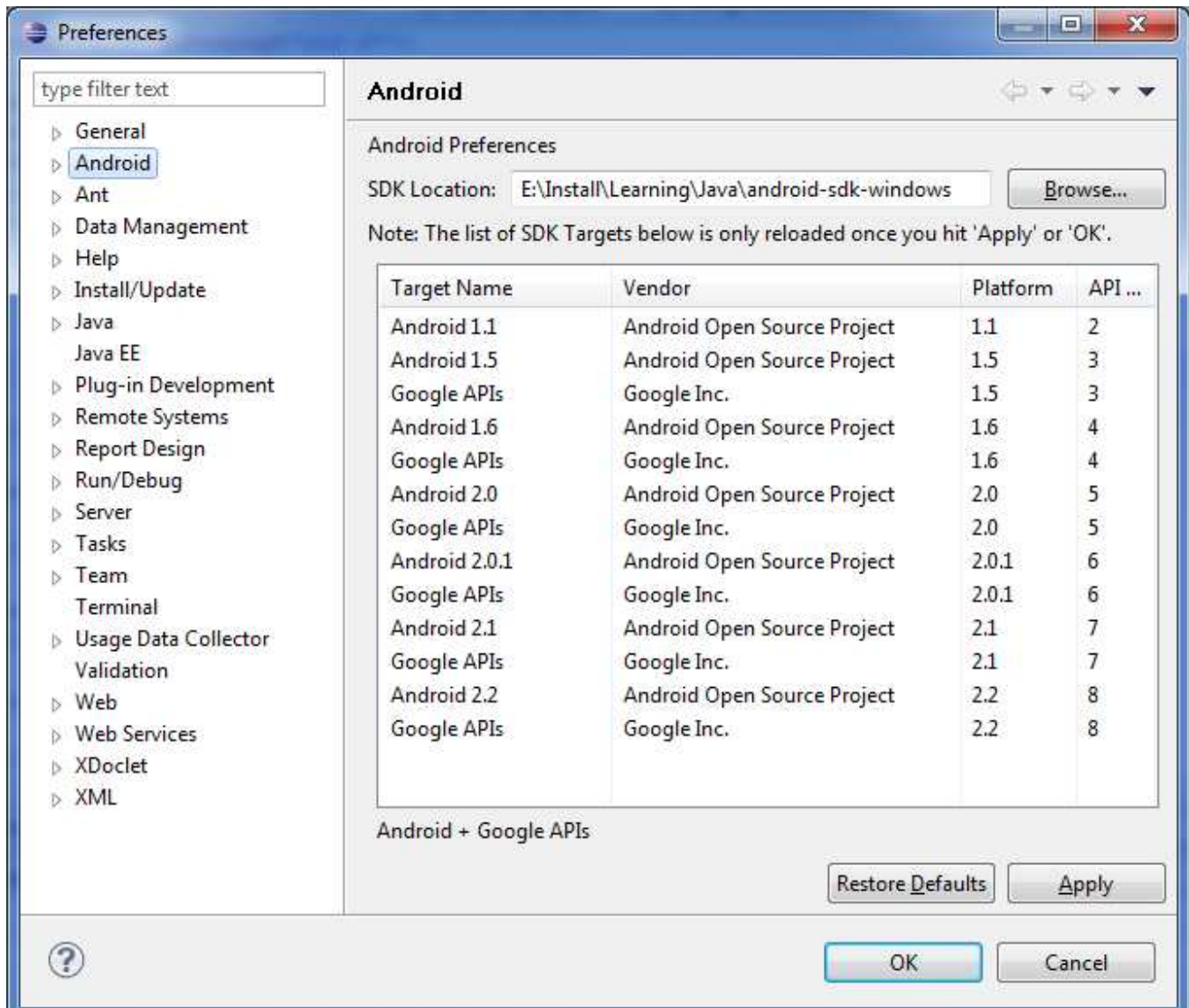
B3: Next, next, Accept, next,...Finish (như Install mọi chương trình bình thường).

B4: Eclipse -> Windows -> Preferences -> Android

Nhấn nút Browse và chỉnh đường dẫn tới thư mục của Android SDK bạn tải lúc trước.

-> Apply

-> OK



3. Android Virtual Device (Emulator):

AVD là máy ảo cho developer phát triển ứng dụng. Để tạo 1 AVD bạn vào Windows -> Android SDK and AVD Manager -> Virtual Devices chọn New.

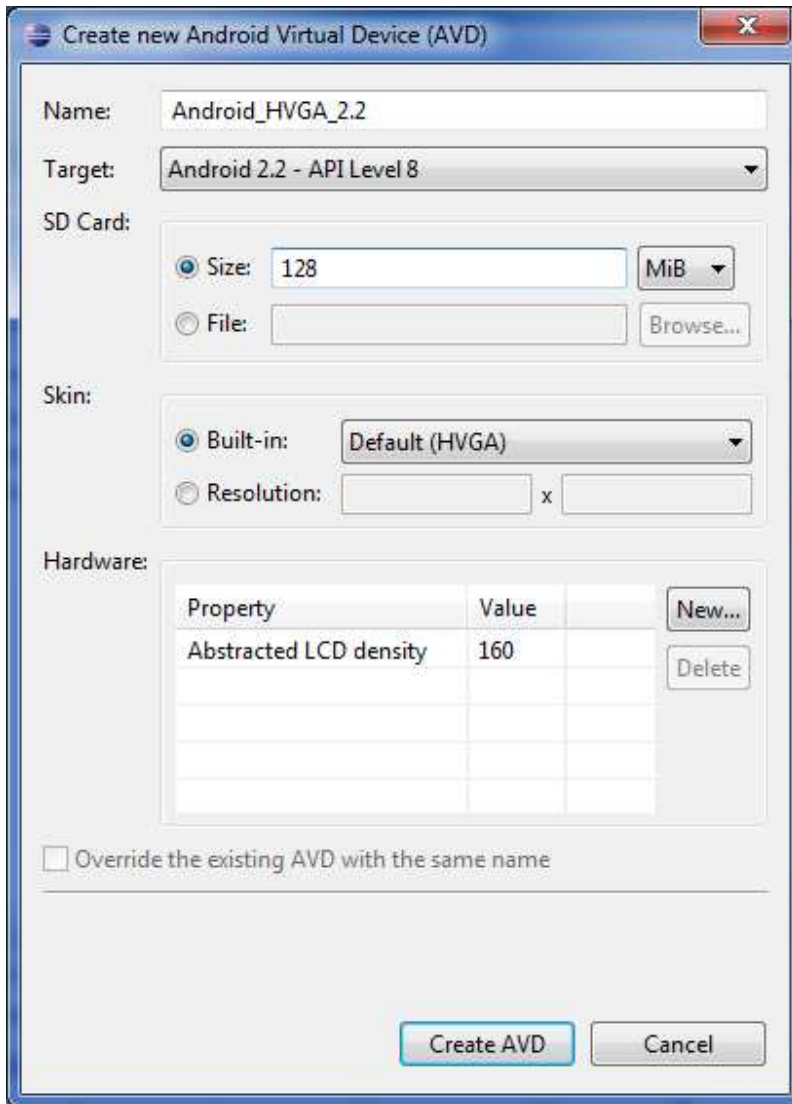
-> Cửa sổ Create new AVD hiện ra, bạn điền thông tin cho AVD bạn muốn:

Name: Tùy ý (nhưng chỉ được sử dụng các ký tự "a-z", "A-Z", ".", "-", nghĩa là cả khoảng trắng cũng ko đc).

Target: Chọn phiên bản hệ điều hành bạn muốn (thường mình tạo một Android 1.6 và một Android 2.2 để test).

SD Card: gõ vào Size SD card ảo cho AVD, hoặc chỉnh tới file đã có sẵn. Nhiều AVD có thể dùng chung 1 Sdcard (chỉ cần tạo lần đầu, các lần sau chỉnh đường dẫn tới file đó).

Skin: có thể để Default (HVGA) hoặc chọn kích cỡ màn hình bạn muốn. Chỉ cần quan tâm tới 3 option: HVGA (phân giải 320-480 như G1, G2, i5700...), QVGA (240-320 như HTC Wildfire...), WVGA854 (480-854 như Milestone, NexusOne...)
-> Create AVD.



II. Một số chức năng cần biết khi lập trình Android với Eclipse:

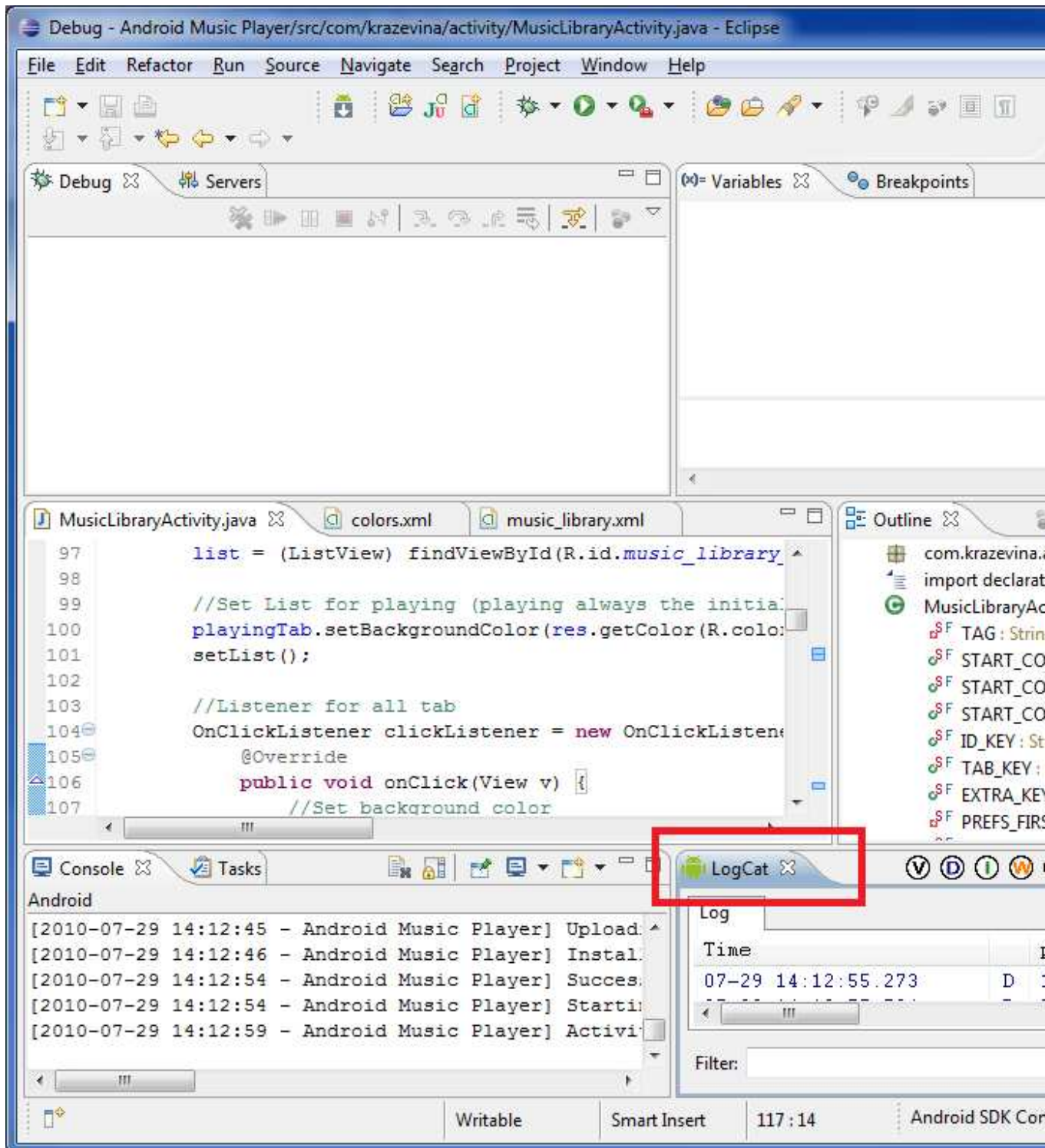
Những điều mình nêu ra dưới đây đều là cơ bản, có thể các pro đã biết hết rồi, nhưng như mình đã nói, bài này là dành cho newbie với hi vọng gia tăng kiến thức cơ bản cùng giảm thời gian tìm kiếm cho các bạn.

1. Debug cho ứng dụng Android:

Nhiều người chuyển từ J2SE hoặc J2ME sang Android sẽ ngạc nhiên vì câu lệnh debug kinh điển **System.out.println()** không còn in ra trên cửa sổ Console nữa. Google đã thay thế nó bằng Logcat, một cửa sổ ghi lại toàn bộ hoạt động của hệ điều hành. Để mở Logcat, trước tiên

các bạn chọn Window -> Open Perspective -> Debug. Nếu ko thấy option Debug thì chọn Other và tìm Debug trong cửa sổ mới hiện ra. Sau đó chọn tab Debug mới xuất hiện ở góc trên bên phải của Eclipse (xem hình). Theo kinh nghiệm của mình thì tốt nhất là Maximize Logcat ra luôn cho dễ quan sát.

Cảm nhận ngày xưa khi mới sử dụng Logcat là rối và khó dùng. Nhưng càng về sau mình càng quen và thấy nó tiện hơn Console nhiều, vì Console chỉ đưa ra thông báo do chúng ta gọi, còn Logcat thì đưa cả luôn những thông báo của hệ điều hành, giúp chúng ta nắm được hệ điều hành đang làm gì, gọi đến cái gì, khởi chạy những gì...

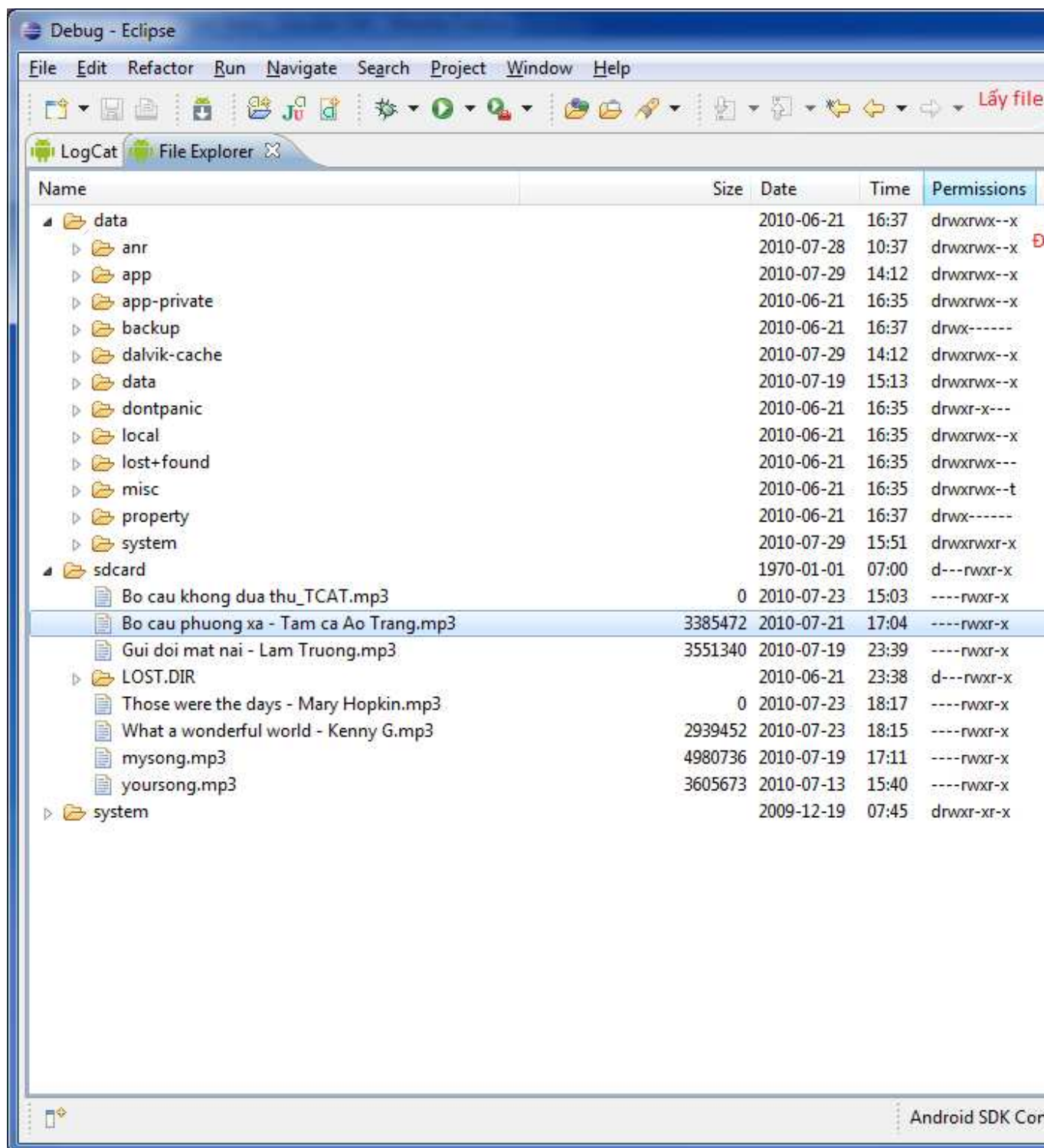


2.File Explorer của Android:

File Explorer là một chức năng hữu ích Google đưa vào giúp chúng ta quản lý file trong sd card

và cả file system data (chỉ quản lý được của emulator, không thể truy nhập system data của thiết bị thật). FE giúp bạn dễ dàng đưa file vào / lấy file ra trong sdcard ảo của emulator, xóa cơ sở dữ liệu của chương trình để khởi tạo lại (only emulator)...

Mở FE bằng cách vào Window -> Show View -> Others -> Android -> File Explorer. Mình để FE trong cửa sổ Debug cho tiện quản lý.

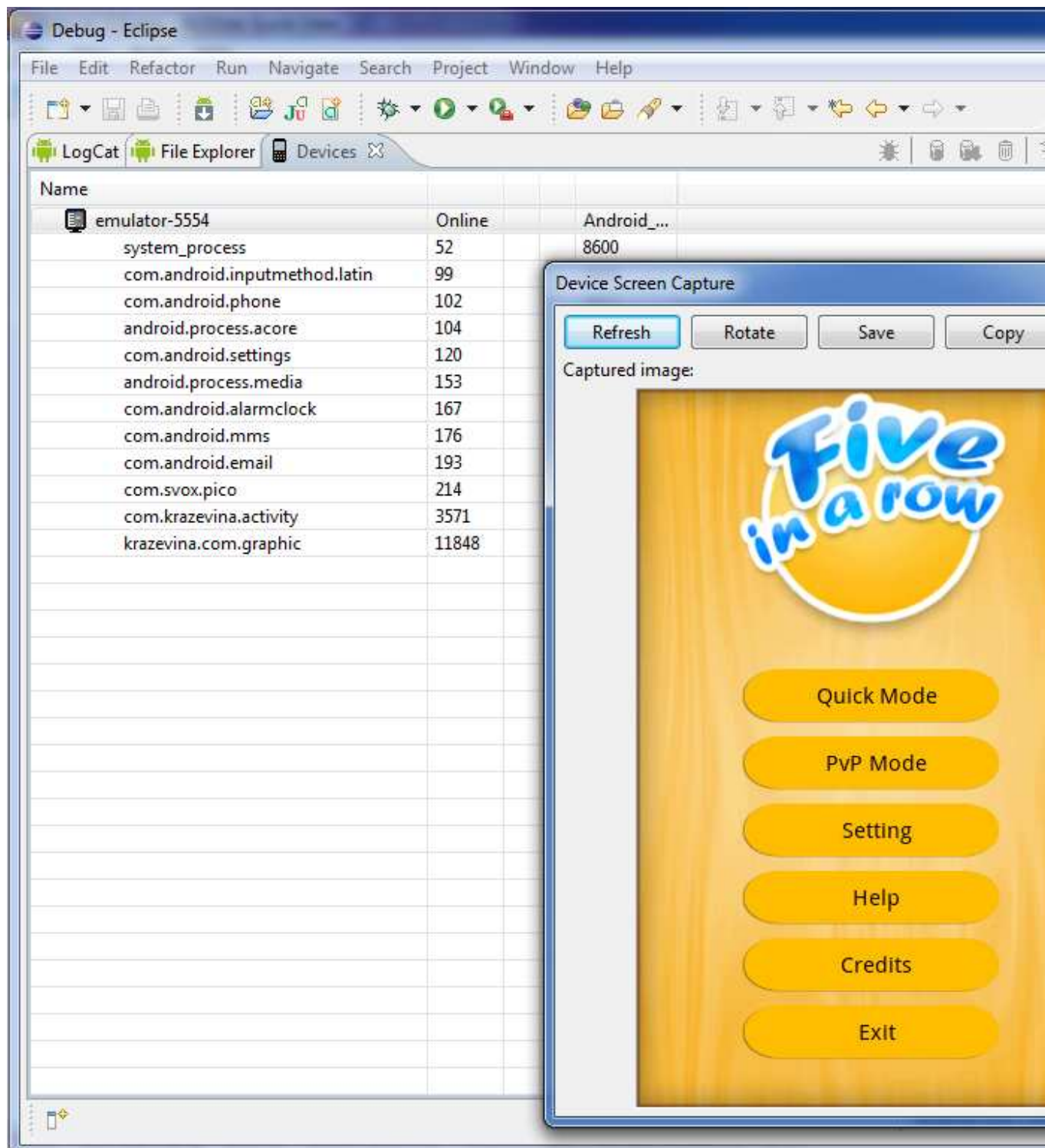


3.Device của Android:

Device cũng là một chức năng hữu ích nữa trong Android giúp bạn quản lý thiết bị ảo cũng như

thật của mình. Mở Device bằng cách vào Window -> Show View -> Device hoặc vào Window -> Show View -> Others -> Android -> Device.

Chức năng mình thường sử dụng nhất của device là Screen Capture, cực kỳ tiện để lấy ảnh minh họa làm thuyết trình hoặc giới thiệu trên Google Market.



III. Một số thủ thuật trong Android và Eclipse:

1. Hiện thị tiếng Việt:

Window -> Preferences -> General -> Workspace.

Trong Workspace phần Text File Encoding chọn Other -> UTF-8.

2. Tự động số code:

Tổ hợp phím kinh điển Ctrl + Space (điều kỳ lạ là khá nhiều người không biết).

3. Comment:

Ctrl + "/": tự động thêm cụm "/*" vào đầu dòng (ko tiện lắm).

Ctrl + Shift + "/": tự động thêm "/* */" vào cụm được bôi đen.

Ctrl + Shift + "\": tự động bỏ "/* */" vào cụm được bôi đen.

4. Override:

Đôi lúc bạn muốn Override phương thức của một lớp có sẵn, ko lẽ chúng ta sẽ đi tìm tên phương thức trên mạng và gõ đúng lại như thế? Nhiều bạn đã gặp lỗi khi định viết lại phương thức này và debug mãi ko ra (vì nó ko phải là lỗi):

Mã:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

}
```

đơn giản vì gõ thiếu chữ s.

Eclipse cung cấp phương tiện giúp ta ghi đè phương thức của lớp cha với Source -> Override/Implement Method, nhưng cực kỳ nhiều người không biết.