CHƯƠNG 2
PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN TẢNG ANDROID

Nội dung chương 2:

1. Giới thiệu tổng quan về Android
2. Các thành pần của Android project (Trong thực hành)
3. Các thành phần trong ứng dụng Android
4. Dòng đời của một ứng dụng Android
5. Các tiến trình trong một ứng dụng Android
6. Cài đặt môi trường phát triển ứng dụng

2.1. GIỚI THIỆU TỔNG QUAN VỀ ANDROID

2.1.1. Quá trình hình thành và phát triển

- Android là tên một nền tảng mở cho thiết bị di động dựa trên nền tảng Linux được phát triển bởi Open Handset Alliance (Google, Samsung, Motorola, HTC, Intel, Qualcomn, Texas Instrument,...), được thành lập năm **2007** và phát triển nhanh chóng. Số thành viên của OHA đã lên đến 84.
- 2008, điện thoại Android đầu tiên ra mắt: T-Mobile G1
- 2010, Google Nexus One - sản phẩm của Google được sản xuất bởi HTC ra mắt đánh dấu bước phát triển mạnh mẽ của Android.
- mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác
- Mỗi phiên bản nâng cấp đều được đặt tên các món ăn tráng miệng. Hiện tại các phiên bản chính của Android bao gồm:

1. Quá trình hình thành và phát triển (2)

- **1.0:** Phiên bản đầu tiên, phát hành ngày 23 tháng 11 năm 2008, có mặt trên HTC Dream (T-Mobile G1).
- **1.1:** Phát hành ngày 09 tháng 2 năm 2009, vá lỗi về bảo mật nhằm hạn chế sự can thiệp vào cấu trúc ROM của máy.
- **1.5 (Cupcake):** Phát hành ngày 30 tháng 4 năm 2009, hỗ trợ bluetooth A2DP và AVRCP, cập nhật về giao diện người dùng.
- **1.6 (Donut), API 4:** Phát hành ngày 15 tháng 9 nă 2009, hỗ trợ tìm kiếm bằng giọng nói Voice Search, công nghệ CDMA/EVDO, màn hình WVGA.
- **2.0/2.1 (Eclair), API 7:** Phát hành ngày 26 tháng 10 năm 2009, tân trang lại giao diện người dùng, giới thiệu HTML5, hỗ trợ Exchange ActiveSync 2.5.
- **2.2 (Froyo), API 8:** Phát hành ngày 20 tháng 5 năm 2010, nâng cấp tốc độ xử lý, giới thiệu engine Chrome V8 JavaScript, hỗ trợ Adobe Flash, thêm tính năng tạo điểm truy cập Wi-Fi.

Quá trình hình thành và phát triển (3)

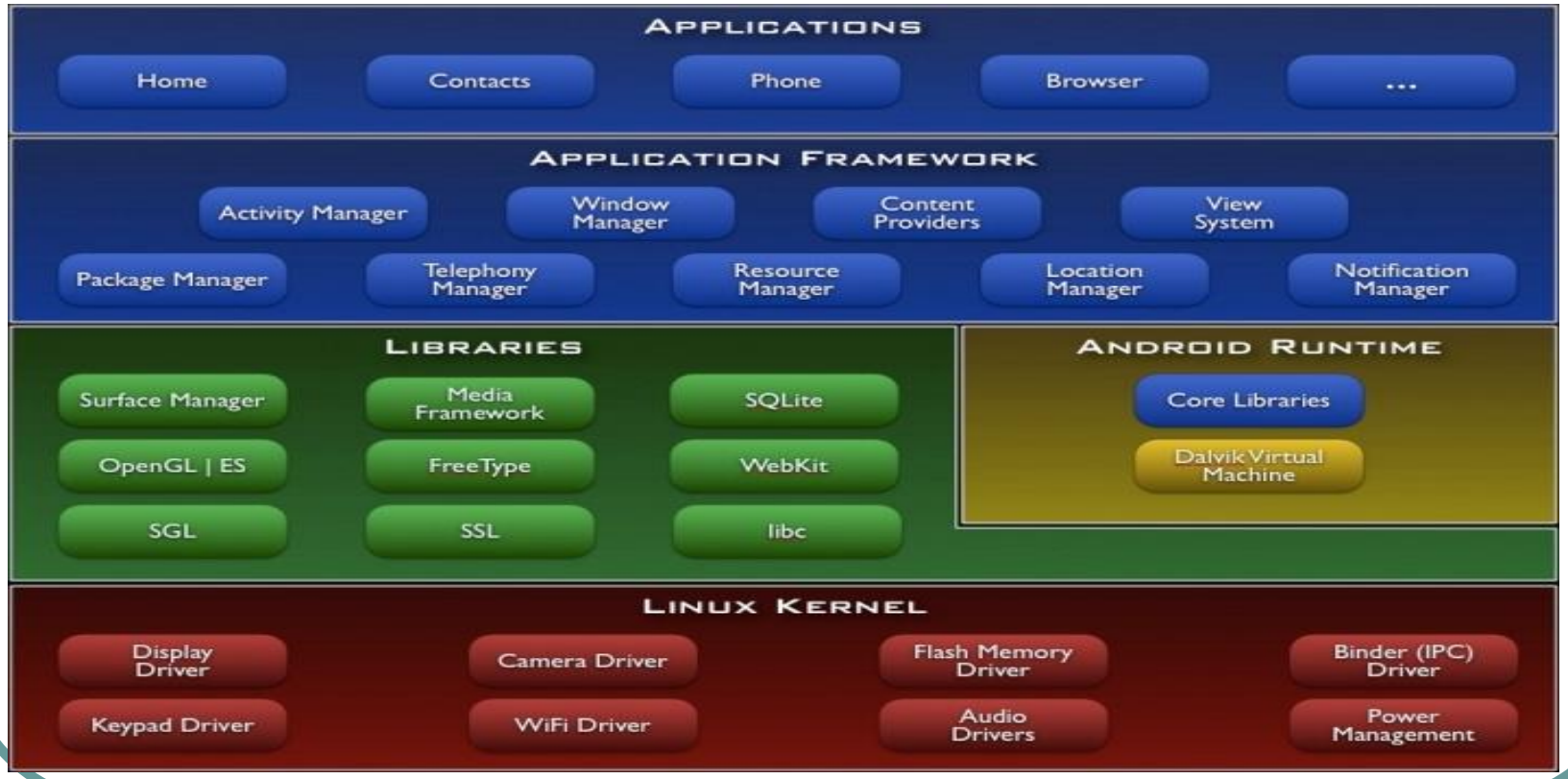
- **2.3 (Gingerbread), API 9,** Phát hành ngày 6 tháng 12 năm 2010,
- **3.1 (Honeycomb), API 12,** tháng 5 năm 2011, hệ điều hành dành riêng cho máy tính bảng.
- **4.0.x (Ice-cream sandwich), API 15,** tháng 12 năm 2011, hệ điều hành mới là sự kết hợp giữa Gingerbread và Honeycomb.
- **Android 4.1 (Jelly Bean), API 16,** tháng 7năm 2012. Với máy tính bảng Nexus 7, với rất nhiều tính năng mới.
- **Android 4.2.x (vẫn là Jelly Bean), API 17,** 13 tháng 11 năm 2012.
- **Android 4.3 (Vẫn là Jelly Bean), API 18,** 25 tháng 7 năm 2013
- **Android 4.4 (KitKat), API 19,** tháng 10 năm 2013.
- **Android 5.0 (Lollipop), API 21,** tháng 11 năm 2014.
- **Android 5.1 (Lollipop_MR1), API 22**, tháng 5 năm 2015. (→ 5.1.1)
- **Android 6.0 (Marshmallow), API 23**, tháng 5 năm 2015. (→ 6.0.1)
- **Android 7.0 (Nougat), API 24,** tháng 8 năm 2016
- Dự kiến **Android 7.1 (Nougat), API ,** tháng 10 năm 2016 ?

2. Ưu và khuyết điểm của Android

- **Q1-2016: trên 2 triệu ứng dụng, 11,1 tỉ lượt tải về → là kho ứng dụng lớn nhất thế giới (đứng thứ 2 là App store của Apple).** [http://thanhnien.vn/cong-nghe/google-play-cham-moc-111-ti-luot-tai-trong-quy-12016-693484.html]
- **Ưu điểm**
  - Tính năng mở
  - Phá bỏ các rào cản
  - Phát triển ứng dụng dễ dàng
  - Gia tăng nhanh về số lượng thiết bị
  - Cộng đồng phát triển rộng lớn
- **Khuyết điểm**
  - Có quá nhiều phiên bản của hệ điều hành
  - Emulator khởi động chậm

3. Kiến trúc của HĐH Android

- Kiến trúc của Android gồm 4 phần:



3. Kiến trúc của HĐH Android(2)

- **Tầng Linux Kernel**

- Đây là nhân của hệ điều hành Android, mọi xử lý của hệ thống đều phải thông qua tầng này. Dựa trên Kernel Linux phiên bản 2.6.
- Nó cung cấp các trình điều khiển các thiết bị phần cứng (driver), quản lý tiến trình, quản lý tài nguyên, bảo mật,...
- Kernel Linux hoạt động như một lớp trừu tượng hóa giữa phần cứng và phần còn lại của phần mềm stack.

3. Kiến trúc của HĐH Android(3)

- **Tầng Library và Android runtime**

- Phần Library : chứa một tập hợp các thư viện được viết từ ngôn ngữ C/C++ để cung cấp cho các thành phần khác của android
- Phần Android runtime: thực chất là một máy ảo Dalvik để chạy các ứng dụng Java.
- Dalvik = máy ảo Java?

3. Kiến trúc của HĐH Android (4)

MÁY ẢO DALVIK

- Dalvik trông giống như máy ảo Java (Java Virtual Machine) nhưng thực tế thì có khác.
- Khi nhà phát triển viết một ứng dụng trong môi trường Java (.java) → được biên dịch sang các bytecode của Java (.class) → chuyển đổi bytecode sang một dạng gọi là dex bytecode (công cụ có tên là dx). "Dex" (Dalvik executable) đóng vai trò như cơ chế ảo thực thi các ứng dụng Android.
- Sự khác biệt giữa JVM và DVM.
- các mã bytecode mà JVM hoạt động là Java bytecode còn DVM hoạt động trên định dạng đặc biệt của nó (định dạng .dex).
- các lớp Java trong chương trình Java SE được biên dịch vào một hay nhiều file (.class) và nén vào file (.jar), sau đó JVM có được các file bytecode từ các file (.class) và file (.jar). Các ứng dụng Android cũng được biên dịch trong ngôn ngữ lập trình Java nhưng sau khi các ứng dụng này được biên dịch thành các file (.class) thì một công cụ dx sẽ biến đổi tất cả các file (.class) thành file (.dex). Từ đó DVM đọc các chỉ dẫn và dữ liệu.

3. Kiến trúc của HĐH Android(5)

- Tầng Application Framework

- Thể hiện nhiều khả năng khác nhau của hệ điều hành Android cho các nhà phát triển ứng dụng để họ có thể sử dụng trong quá trình phát triển ứng dụng.

3. Kiến trúc của HĐH Android(6)

- **Tầng Applications**

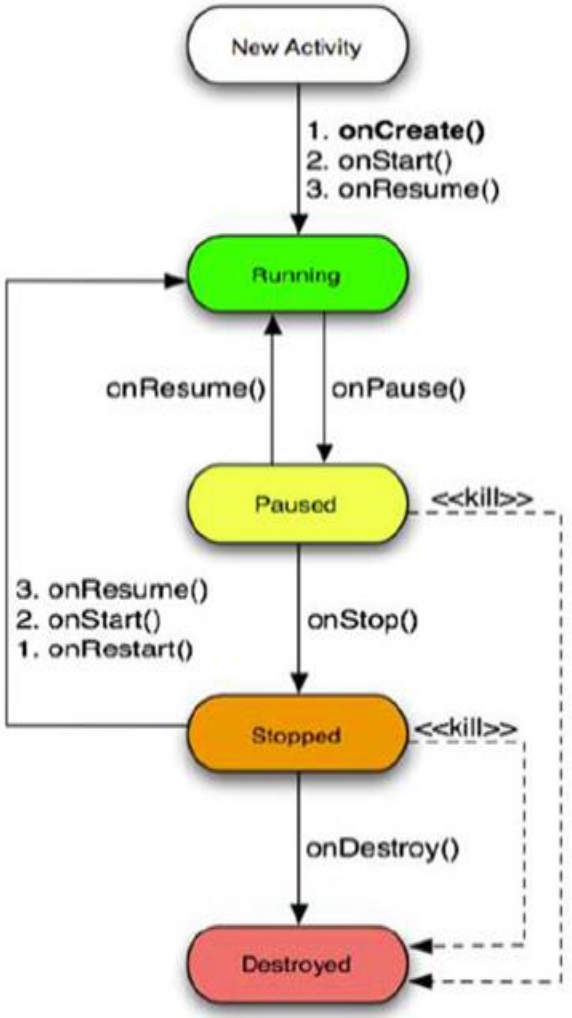
- Là hệ thống phần mềm ứng dụng chạy trên Android, viết bằng Java, chạy trên máy ảo Dalvik Virtual Machine, có thể chạy nền (background) hoặc tích cực (activity).
- Gồm các ứng dụng cơ bản cơ bản được nhà sản xuất tích hợp sẵn trên thiết bị Android (như gọi điện thoại, danh bạ, tin nhắn, email, trình duyệt web, lịch, chụp ảnh, quay phim, thu âm, …), cũng như các ứng dụng được tải về.

CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

- Việc hiểu các thành phần (component) tạo nên một ứng dụng Android là điều kiện tiên quyết cho việc lập trình.

1. **Activity:** là thành phần cơ bản của ứng dụng. Khi khởi động ứng dụng thì bao giờ cũng có 1 Activity chính được gọi, hiển thị giao diện của ứng dụng cho phép người dùng tương tác. Một ứng dụng có thể có một hoặc nhiều Activity.

- Một activity có các trạng thái chính: Active (Running), Paused, Stopped. Khi chuyển giữa các trạng thái, ứng dụng sẽ gọi các hàm tương ứng.
- ***Vòng đời của Activity***
  - ***Entire lifetime:*** *Từ* onCreate() →onDestroy();
  - ***Visible lifetime:*** Từ onStart() → onStop;
  - ***Foreground lifetime:*** Từ onResume() → onPause().
- Cần phải viết lại (Override) phương thức onCreate()
- Các Activity bổ sung thêm phải được khai báo trong tập tin AndroidManifest.xml, để có thể gọi các Activity này thì cách phổ biến là sử dụng thành phần Intent.



CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="buoi1.bai3"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="15" />
    <application android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
        <activity android:name=".Activity1“
```

```
// Gọi Activity chính bằng thành phần Intent.
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Activity2" >
        </activity>
    </application>
</manifest>
```

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

2. Service

- là thành phần chạy ẩn (chạy ngầm), không có giao diện, vì vậy người dùng không thể nhận ra nó trong ứng dụng. Service dùng để cập nhật dữ liệu, đưa ra các cảnh báo (Notification). → Mỗi Service đều được mở rộng từ lớp cơ sở là Service trong gói android.app.
- Service cũng có vòng đời (từ khi được khởi chạy cho tới khi kết thúc) bao gồm nhiều trạng thái. Việc chuyển đổi từ trạng thái này sang trạng thái khác được hỗ trợ bởi các phương thức tương ứng. Ví dụ: phương thức Context.startService() dùng để khởi chạy một Service và nó được dừng bằng phương thức Context.stopService(),...

3. Content Provider

- Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng.
- Dữ liệu thường được lưu trữ ở tập tin hệ thống, hoặc trong một SQLite database. Ví dụ: danh bạ, call log, cấu hình cài đặt..., trên điện thoại là dữ liệu dưới dạng Content Provider.

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

4. Intent

- là công cụ truyền tải các thông báo được sử dụng để gửi các thông báo đi nhằm khởi tạo một Activity hay Service để thực hiện công việc mong muốn.
- Intent là một cơ cấu cho phép truyền thông điệp giữa các thành phần của một ứng dụng và giữa các ứng dụng với nhau. Ví dụ, khi mở một trang web, ta gửi một intent đi để gọi một activity mới hiển thị trang web đó.
- **Các thuộc tính của Intent:**

Thuộc tính chính	Thuộc tính phụ
action -tên (string) của action mà Intent sẽ yêu cầu thực hiện -có thể là action được Android định nghĩa sẵn (built-in standard action) hoặc do người lập trình tự định nghĩa	**category** -thông tin về nhóm của action
	type -định dạng kiểu dữ liệu (chuẩn MIME) -thường được tự động xác định
data -dữ liệu mà Activity được gọi sẽ xử lý -định dạng Uri (thông qua hàm Uri.parse(data))	**component** -chỉ định cụ thể lớp sẽ thực thi Activity -khi được xác định, các thuộc tính khác trở thành không bắt buộc (optional)
	extras -chứa tất cả các cặp (key,value) do ứng dụng thêm vào để truyền qua Intent (cấu trúc Bundle)
http://developer.android.com/reference/android/content/Intent.html	

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

- **action:** là hành động được thực hiện, ví dụ : ACTION_VIEW, ACTION_MAIN.

Built-in Standard Actions	
ACTION_MAIN	ACTION_ANSWER
ACTION_VIEW	ACTION_INSERT
ACTION_ATTACH_DATA	ACTION_DELETE
ACTION_EDIT	ACTION_RUN
ACTION_PICK	ACTION_SYNC
ACTION_CHOOSER	ACTION_PICK_ACTIVITY
ACTION_GET_CONTENT	ACTION_SEARCH
ACTION_DIAL	ACTION_WEB_SEARCH
ACTION_CALL	ACTION_FACTORY_TEST
ACTION_SEND	ACTION_SENDTO
Built-in Standard Broadcast Actions	
ACTION_TIME_TICK	ACTION_PACKAGE_RESTA
ACTION_TIME_CHANGED	ACTION_PACKAGE_DATA
ACTION_TIMEZONE_CHANGED	ACTION_UID_REMOVED
ACTION_BOOT_COMPLETED	ACTION_BATTERY_CHANG
ACTION_PACKAGE_ADDED	ACTION_POWER_CONNECT
ACTION_PACKAGE_CHANGED	ACTION_POWER_DISCONN
ACTION_PACKAGE_REMOVED	ACTION_SHUTDOWN

Hình 2-13: Action được dựng sẵn

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

- **data:** là dữ liệu sẽ được xử lý trong action, thường được diễn tả là một Uri (Uniform Resource Identifier). Ví dụ:
  - ACTION_VIEW content://contacts/people/1 - Hiển thị thông tin về người với mã danh sách 1.
  - ACTION_DIAL content://contacts/people/1 - Hiển thị màn hình gọi đến người với mã danh sách 1.
  - ACTION_DIAL tel:1800 - Hiển thị màn hình gọi với số gọi là 1800.
- Ngoài ra còn có 1 số thuộc tính mà ta có thể bổ sung vào Intent:
  - **category:** bổ sung thêm thông tin cho action của Intent. VD: CATEGORY_LAUNCHER thông báo sẽ thêm vào Launcher như là một ứng dụng top-level;
  - **type:** chỉ rõ kiểu dữ liệu;
  - **component:** chỉ rõ thành phần sẽ nhận và xử lý intent. Khi thuộc tính này được xác định thì các thuộc tính khác sẽ trở thành thuộc tính phụ;
  - **extras:** mang theo đối tượng Bundle chứa các giá trị bổ sung.

  Ví dụ: ACTION_MAIN: khởi chạy một Activity; CATEGORY_HOME: trở về màn hình Home của Android (khi bấm nút Home của di động); ACTION_SEND: mở một Activity cho phép gửi dữ liệu lấy từ data URI, kiểu của dữ liệu xác định trong thuộc tính type.

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

❖ **Intent được chia làm 2 loại**

- **Explicit Intents**: là intent đã được xác định thuộc tính component, nghĩa là đã chỉ rõ thành phần sẽ nhận và xử lý intent. Thông thường intent dạng này sẽ không bổ sung thêm các thuộc tính khác như action, data. Explicit Intent thường được sử dụng để khởi chạy các activity trong cùng một ứng dụng.
- **Implicit Intents:** Intent không chỉ rõ component được xử lý, thay vào đó nó bổ sung thông tin trong các thuộc tính. Khi intent được gửi đi, hệ thống sẽ dựa vào những thông tin này để quyết định component nào thích hợp nhất để xử lý nó.
- **Intent Filter:** Activity, Service và BroadCast Receiver sử dụng Intent Filter để thông báo cho hệ thống biết các dạng Implicit Intent mà nó có thể xử lý. (bộ lọc Intent, chỉ cho những Intent được phép đi qua nó).
- Mặc định mỗi ứng dụng đều đã khai báo Intent Filter, nếu muốn bổ sung thêm Intent Filter khác thì cũng phải khai báo thêm trong AndroidManifest.xml.

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

Ví dụ:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.vidu"
……..
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
<intent-filter>
<action android:name="android.intent.action.CALL_BUTTON" />
<category android:name="android.intent.category.DEFAULT" />
</intent-filter>
……..
</manifest>
```

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

5. Broadcast Receiver (BR)

- BR có chức năng dùng để nhận các sự kiện mà các ứng dụng hoặc hệ thống phát đi. Ví dụ: khi viết một chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần một BR để nhận biết các Intent là các cuộc gọi tới.
- BR là thành phần mà không làm gì khác ngoài việc nhận các thông báo được phát đến (broadcast). Một ứng dụng có thể có nhiều broadcast receiver để đáp lại những thông báo được phát đến. Một BR không có giao diện nhưng nó có thể thực hiện gọi một Activity hay là sử dụng NotificationManager để thông báo cho người dùng.
- Một số broadcast thường gặp như báo hệ thống khởi động hoàn tất, báo pin có sự thay đổi, thông báo tin nhắn tới, thông báo cắm/rút thẻ nhớ,…
- **Có 2 cách phát-nhận đó là:**

- **Không có thứ tự:** BR đủ điều kiện thì sẽ được nhận, không phân biệt và tách rời nhau.
- **Có thứ tự:** receiver đăng ký ưu tiên hơn sẽ được nhận trước, và có thể truyền thêm thông tin xử lý cho các receiver sau.

2.3. CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID

❖ **Có hai lớp broadcast chính:**

- **Normal broadcast** (được gửi bằng Context.sendBroadcast) hoàn toàn không đồng bộ (asynchronous). Tất cả các receiver của broadcast được chạy theo thứ tự không xác định, thường là chạy cùng lúc.
- **Ordered broadcast** (được gửi bằng Context.sendOrderedBroadcast) được gửi lần lượt đến cho từng receiver một. Thứ tự thực thi của các receiver được kiểm soát bởi thuộc tính android:priority của intent-filter tương ứng; các receiver có cùng priority (độ ưu tiên) sẽ được chạy theo thứ tự bất kỳ.
- Khai báo trong thẻ <application> của tập tin AndroidManifest.xml:

```
<receiver android:name="BroadcastReceiver">
<intent-filter>
<action android:name = "vidu.BR" />
</intent-filter>
</receiver>
```

6. Notification dùng để đưa ra các cảnh báo mà không làm cho các Activity phải ngừng hoạt động.

➢ Trong các thành phần trên thì **Activity, Service, Broadcast Receiver** và **Content Provider** là 4 thành phần chính cấu thành nên ứng dụng Android.

2.4. Dòng đời của một ứng dụng Android

- Trong một ứng dụng Android có chứa nhiều thành phần và mỗi thành phần đều có một chu trình sống riêng.
- Ứng dụng chỉ được gọi là kết thúc khi tất cả các thành phần trong ứng dụng kết thúc.
- Activity là một thành phần cho phép người dùng giao tiếp với ứng dụng.
- Khi tất cả các Activity kết thúc và người dùng không còn giao tiếp được với ứng dụng nữa nhưng không có nghĩa là ứng dụng đã kết thúc. Bởi vì ngoài Activity là thành phần có khả năng tương tác người dùng thì còn có các thành phần không có khả năng tương tác với người dùng như là Service, BroadcastReceiver,...
- Nếu một Activity được tạm dừng hoặc dừng hẳn, hệ thống có thể bỏ thông tin của nó từ vùng nhớ bởi việc gọi hàm finish().

2.6. Các tiến trình

- Android có cơ chế quản lý các tiến trình (process) theo chế độ ưu tiên (priority). Các tiến trình có độ ưu tiên thấp sẽ bị giải phóng mà không hề có cảnh báo nhằm đảm bảo tài nguyên.
- Foreground process: là tiến trình của ứng dụng hiện thời đang được người dùng tương tác.
- Visible process: là tiến trình của ứng dụng mà Activity đang hiển thị đối với người dùng (phương thức onPause() của Activity được gọi).
- Service process: là Service đang hoạt động (running).
- Background process: là tiến trình của ứng dụng mà các Activity của nó không hiển thị với người dùng (phương thức onStoped() của Activity được gọi).
- Empty process: tiến trình không có bất cứ thành phần nào hoạt động.
- Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động đóng tiến trình (kill process), trước tiên là các Empty process.

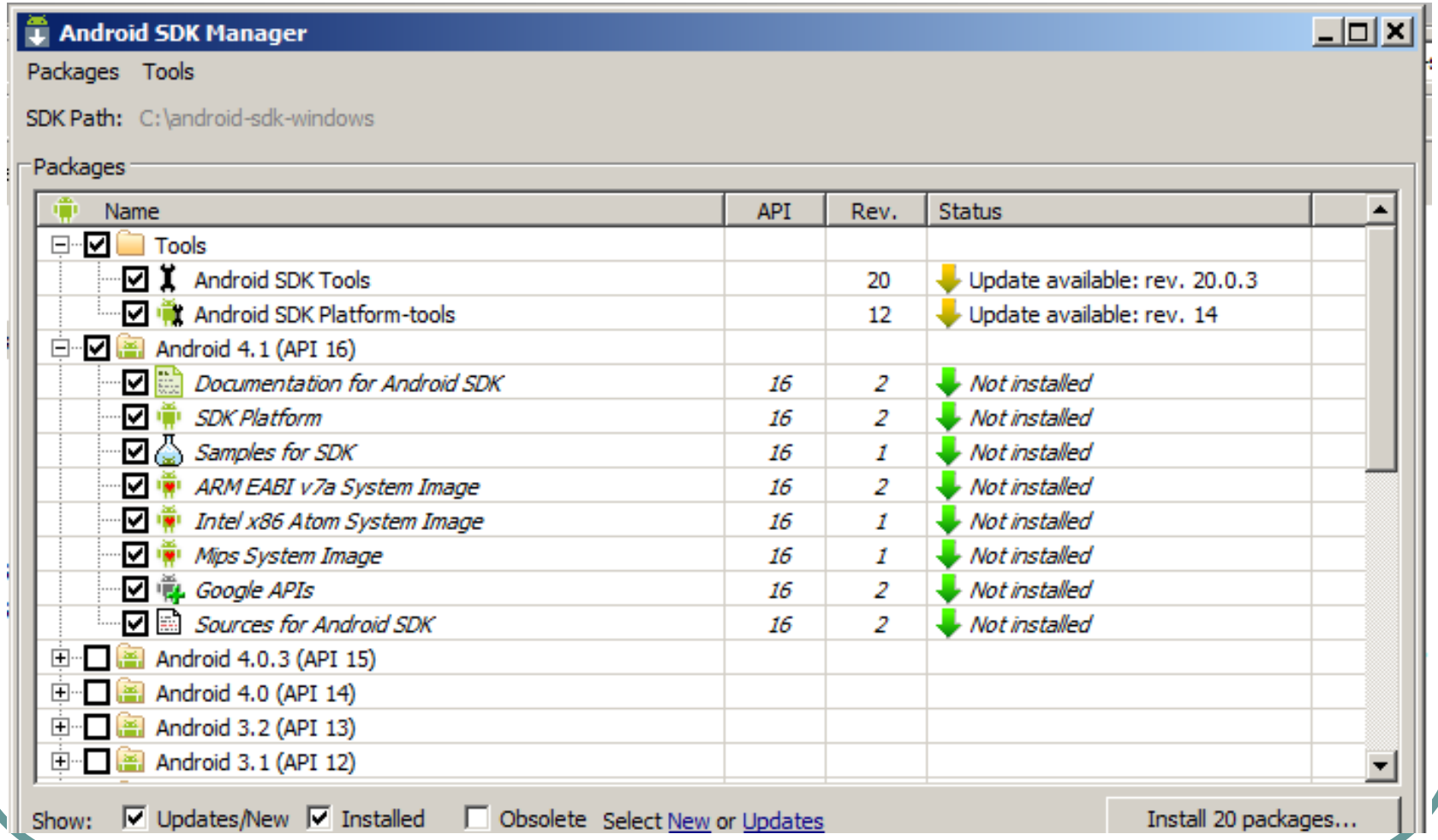
Phần II: Hướng dẫn cài đặt Android trên Eclipse

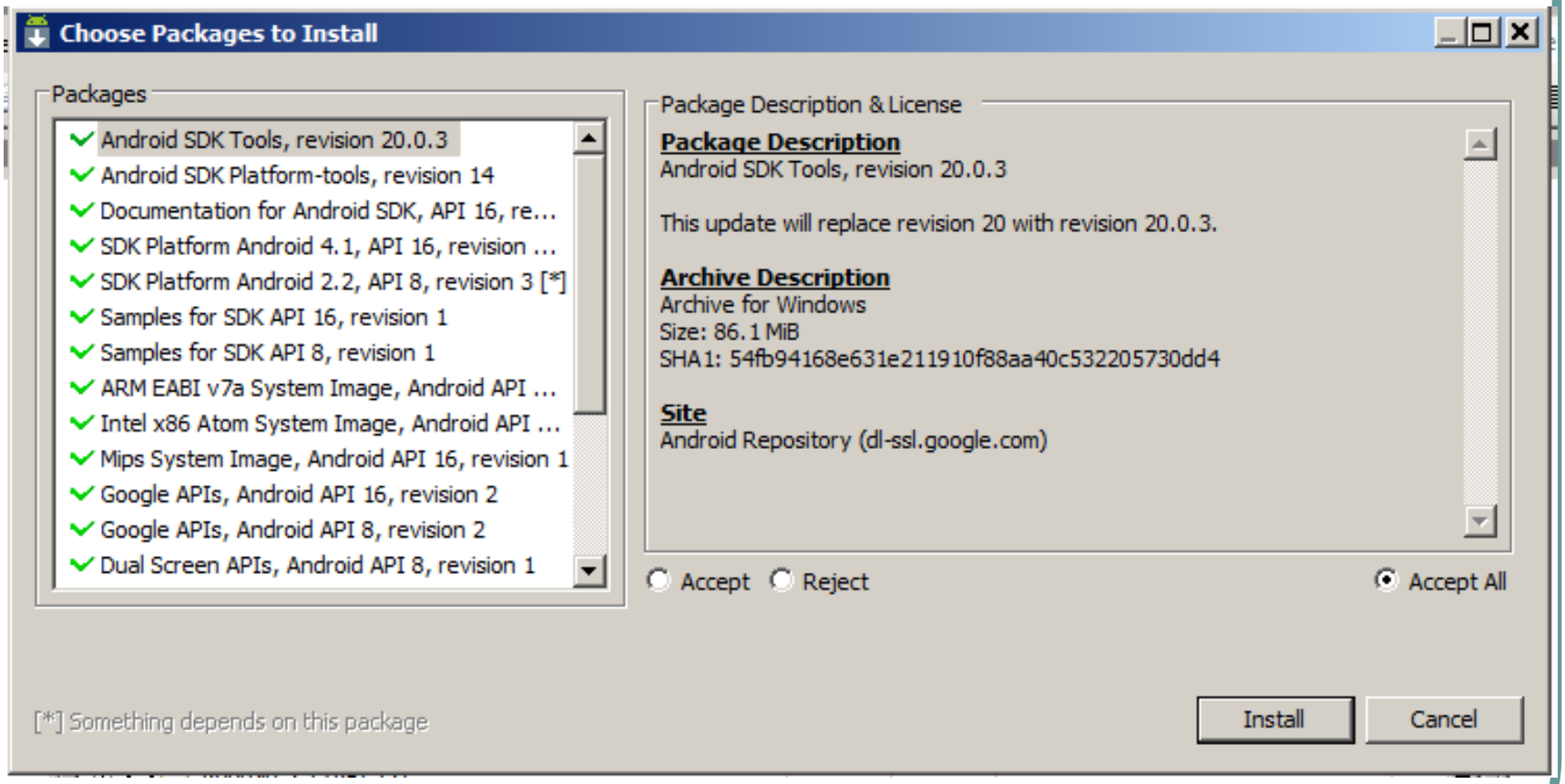
1. Chuẩn bị và cài đặt

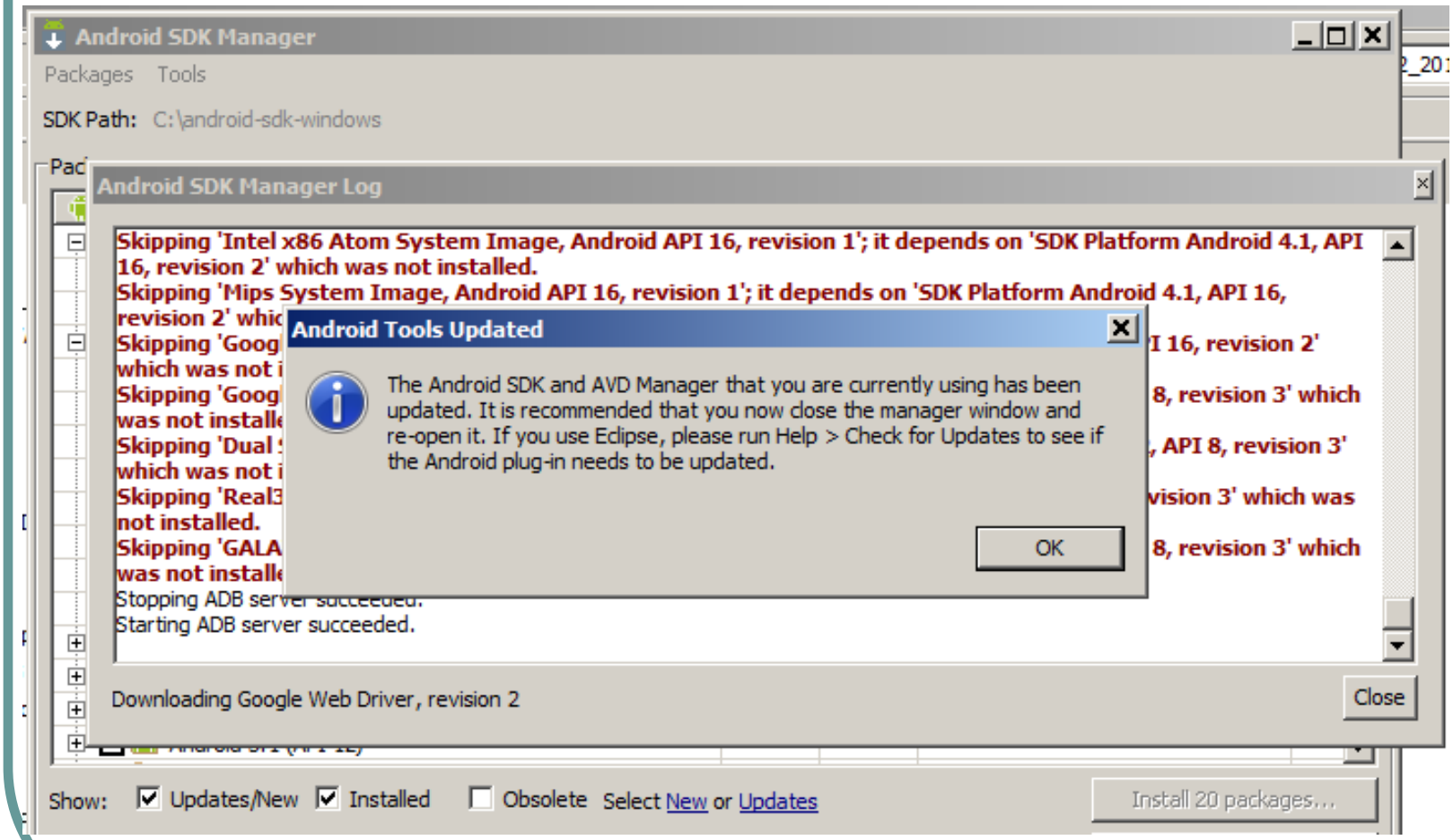
- Máy tính cài đặt một trong các hệ điều hành là Microsoft Window,Linux, MacOs. Phải có kết nối mạng trong quá trình cài đặt.
- JDK 5 hoặc JDK 6. Có thể download miễn phí tại: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Eclipse. Download miễn phí (Nên chọn Eclipse for Java Developers): http://www.eclipse.org/downloads/
- Android software development kit download miễn phí từ: http://developer.android.com/sdk/index.html.
- Thứ tự cài đặt:
  1) jdk
  2) Eclipse
  3) Giải nén tập tin android-sdk vào ổ đĩa C (gồm: add-ons, platforms, tools,SDK Manager, AVD Manager, SDK Readme).
  4) Cài đặt androis SDK (chạy tập tin SDK Manager)
  5) Cập nhật biến môi trường.
  6) Cài đặt plugin cho Eclipse.
  7) Tạo thiết bị giả lập (chạy tập tin AVD Manager).

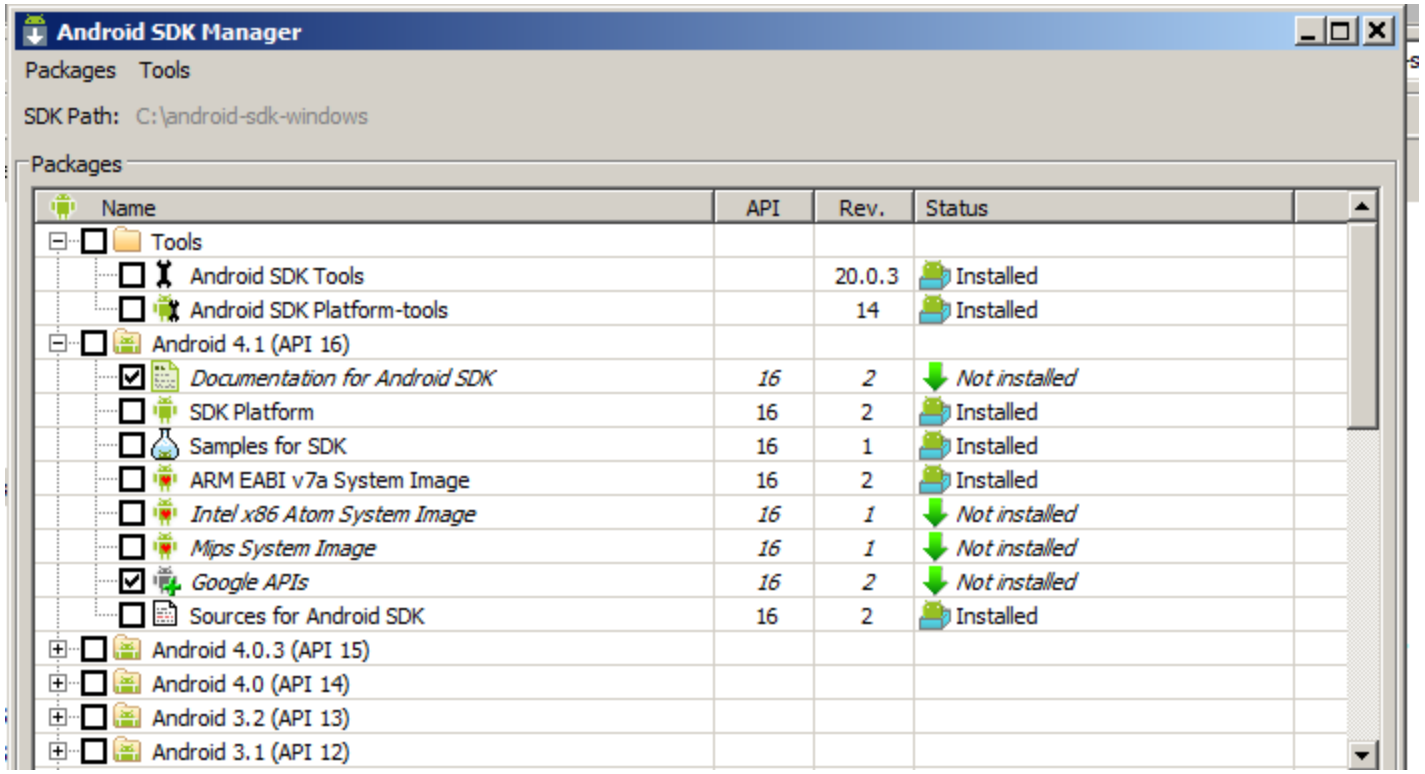
2. Cài Đặt Android SDK

- Chạy tập tin **SDK Manager.exe** → Cửa sổ **Android SDK Manager** hiện ra, nó sẽ tự động tìm kiếm các package → đánh dấu chọn những package cần thiết (Tools, các phiên bản Android, các Extras - Lưu ý những phiên bản có kèm add-on Google APIs) → Nhắp chuột vào **Install xx package** để tiếp tục. → Trên cửa sổ **Choose Packages to Install** chọn **Accept All**. → Cửa sổ **Android SDK Manager Log** và **Android Tools Updated** xuất hiện → Nhắp **OK** và tiếp tục chờ download → Tiếp tục với các gói khác, nếu chưa đủ. (khi mở lại cửa sổ này thì những gói đã được tải về và cài đặt thành công thì cột Status sẽ hiển thị dòng chữ Installed)


Android SDK Manager
Packages Tools
SDK Path: C:\android-sdk-windows
Packages
Name API Rev. Status
Tools
Android SDK Tools 20 Update available: rev. 20.0.3
Android SDK Platform-tools 12 Update available: rev. 14
Android 4.1 (API 16)
Documentation for Android SDK 16 2 Not installed
SDK Platform 16 2 Not installed
Samples for SDK 16 1 Not installed
ARM EABI v7a System Image 16 2 Not installed
Intel x86 Atom System Image 16 1 Not installed
Mips System Image 16 1 Not installed
Google APIs 16 2 Not installed
Sources for Android SDK 16 2 Not installed
Android 4.0.3 (API 15)
Android 4.0 (API 14)
Android 3.2 (API 13)
Android 3.1 (API 12)
Show: Updates/New Installed Obsolete Select New or Updates
Install 20 packages...

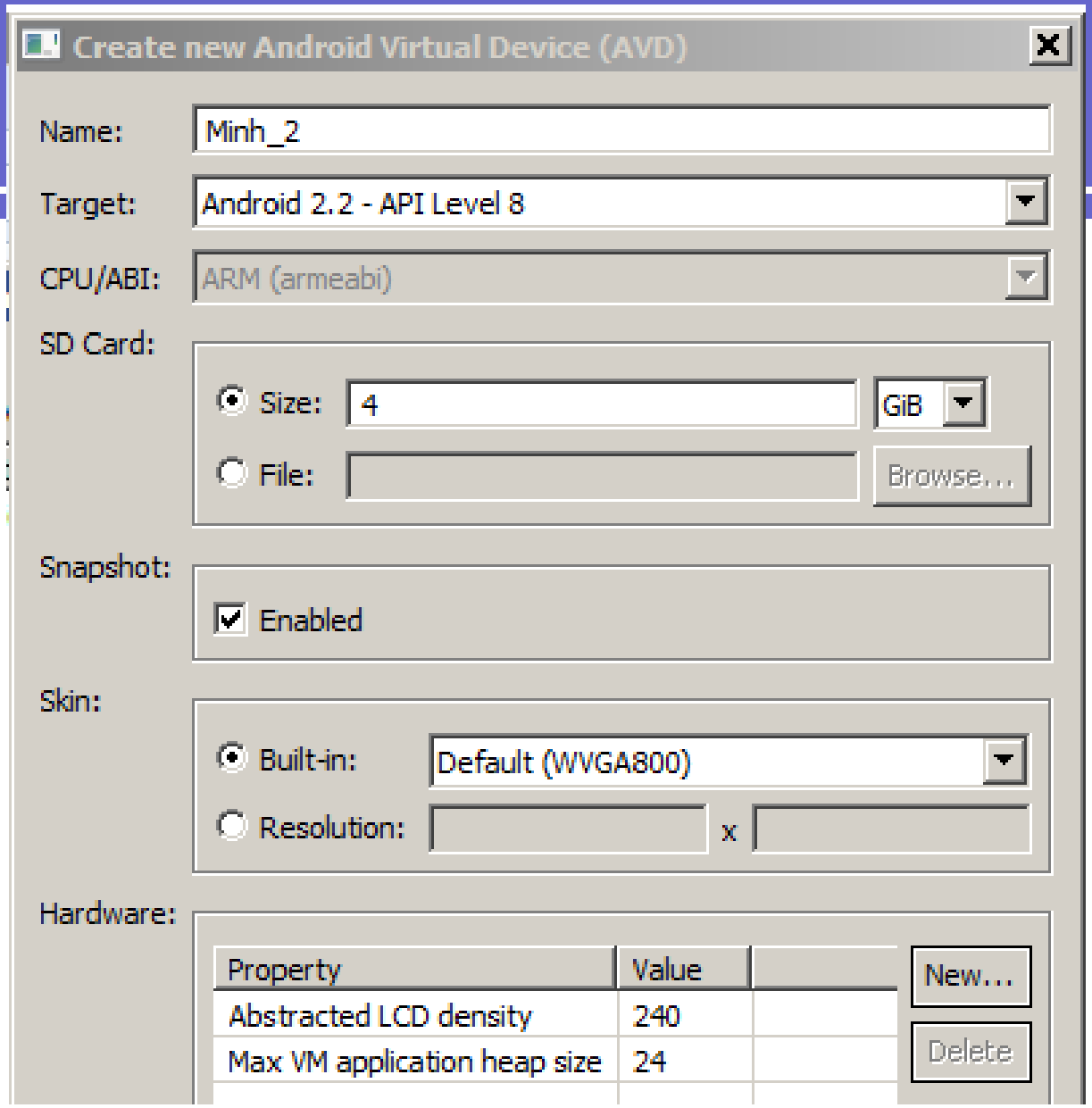

Choose Packages to Install
Packages
Android SDK Tools, revision 20.0.3
Android SDK Platform-tools, revision 14
Documentation for Android SDK, API 16, re...
SDK Platform Android 4.1, API 16, revision ...
SDK Platform Android 2.2, API 8, revision 3 [*]
Samples for SDK API 16, revision 1
Samples for SDK API 8, revision 1
ARM EABI v7a System Image, Android API ...
Intel x86 Atom System Image, Android API ...
Mips System Image, Android API 16, revision 1
Google APIs, Android API 16, revision 2
Google APIs, Android API 8, revision 2
Dual Screen APIs, Android API 8, revision 1
Package Description & License
Package Description
Android SDK Tools, revision 20.0.3
This update will replace revision 20 with revision 20.0.3.
Archive Description
Archive for Windows
Size: 86.1 MiB
SHA1: 54fb94168e631e211910f88aa40c532205730dd4
Site
Android Repository (dl-ssl.google.com)
Accept
Reject
Accept All
[*] Something depends on this package
Install
Cancel

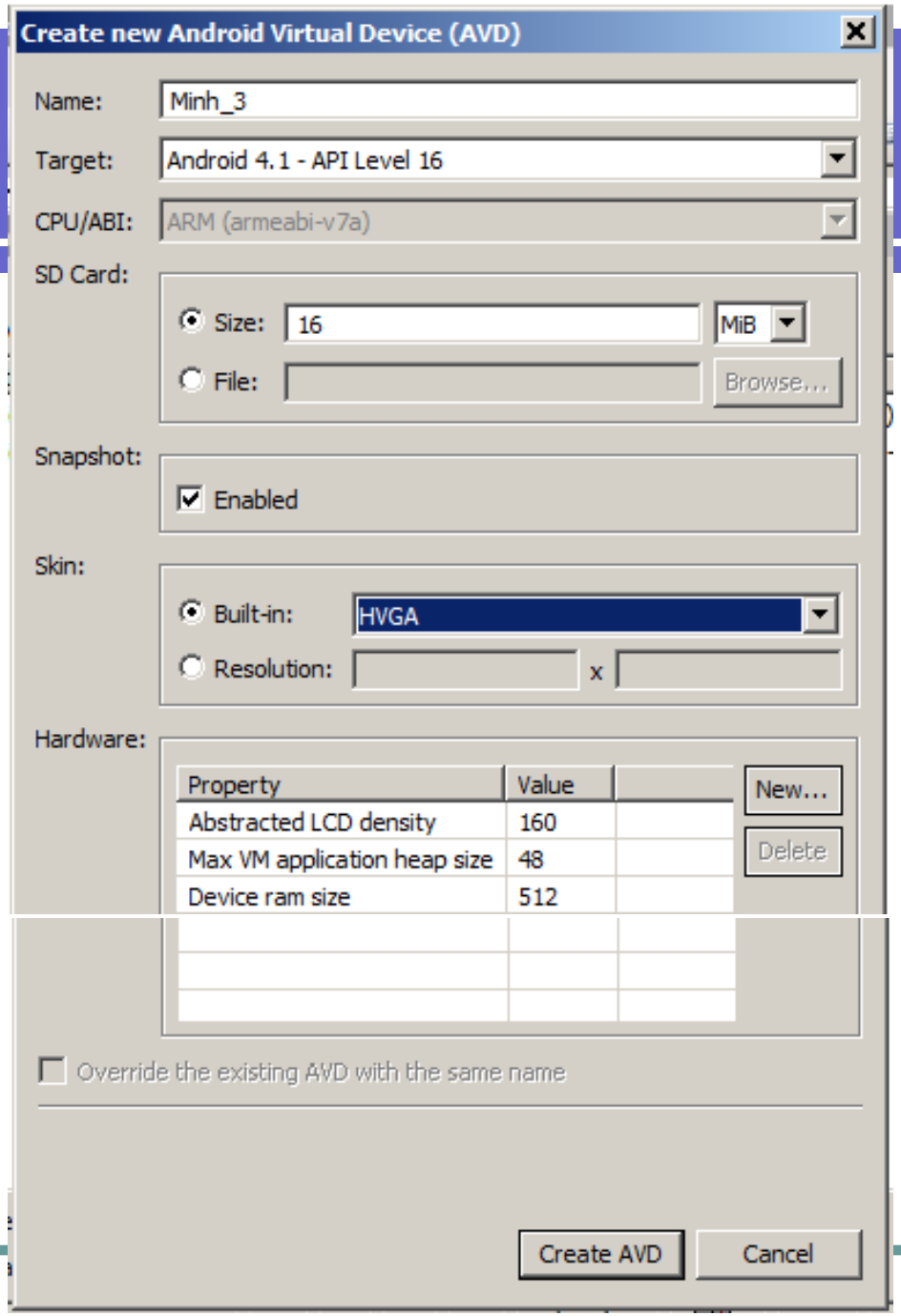

Android SDK Manager
Packages Tools
SDK Path: C:\android-sdk-windows
Android SDK Manager Log
Skipping 'Intel x86 Atom System Image, Android API 16, revision 1'; it depends on 'SDK Platform Android 4.1, API 16, revision 2' which was not installed.
Skipping 'Mips System Image, Android API 16, revision 1'; it depends on 'SDK Platform Android 4.1, API 16, revision 2' whic
Skipping 'Goog
which was not i
Skipping 'Goog
was not installe
Skipping 'Dual
which was not i
Skipping 'Real3
not installed.
Skipping 'GALA
was not installe
Stopping ADB server succeeded.
Starting ADB server succeeded.
Android Tools Updated
The Android SDK and AVD Manager that you are currently using has been updated. It is recommended that you now close the manager window and re-open it. If you use Eclipse, please run Help > Check for Updates to see if the Android plug-in needs to be updated.
OK
Downloading Google Web Driver, revision 2
Close
Show: Updates/New Installed Obsolete Select New or Updates
Install 20 packages...

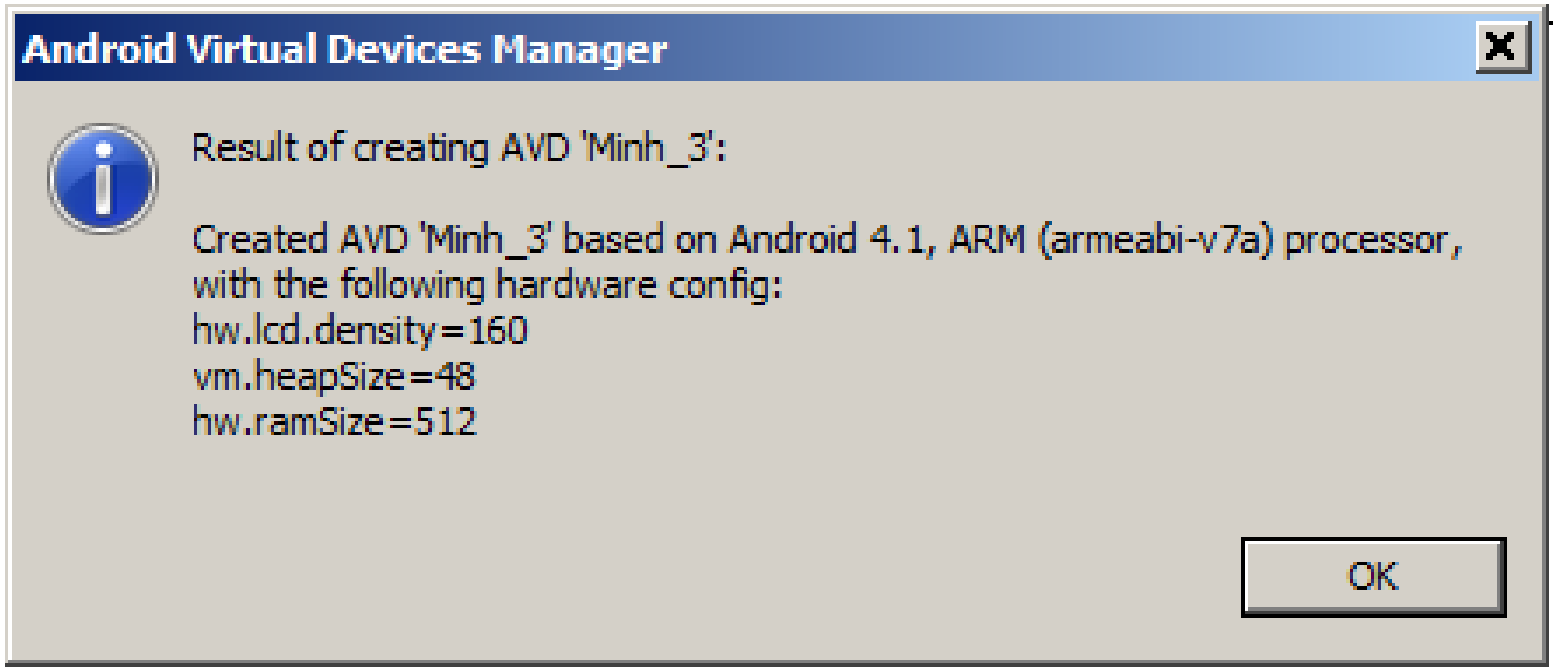

Android SDK Manager
Packages Tools
SDK Path: C:\android-sdk-windows
Packages
Name API Rev. Status
Tools
Android SDK Tools 20.0.3 Installed
Android SDK Platform-tools 14 Installed
Android 4.1 (API 16)
Documentation for Android SDK 16 2 Not installed
SDK Platform 16 2 Installed
Samples for SDK 16 1 Installed
ARM EABI v7a System Image 16 2 Installed
Intel x86 Atom System Image 16 1 Not installed
Mips System Image 16 1 Not installed
Google APIs 16 2 Not installed
Sources for Android SDK 16 2 Installed
Android 4.0.3 (API 15)
Android 4.0 (API 14)
Android 3.2 (API 13)
Android 3.1 (API 12)

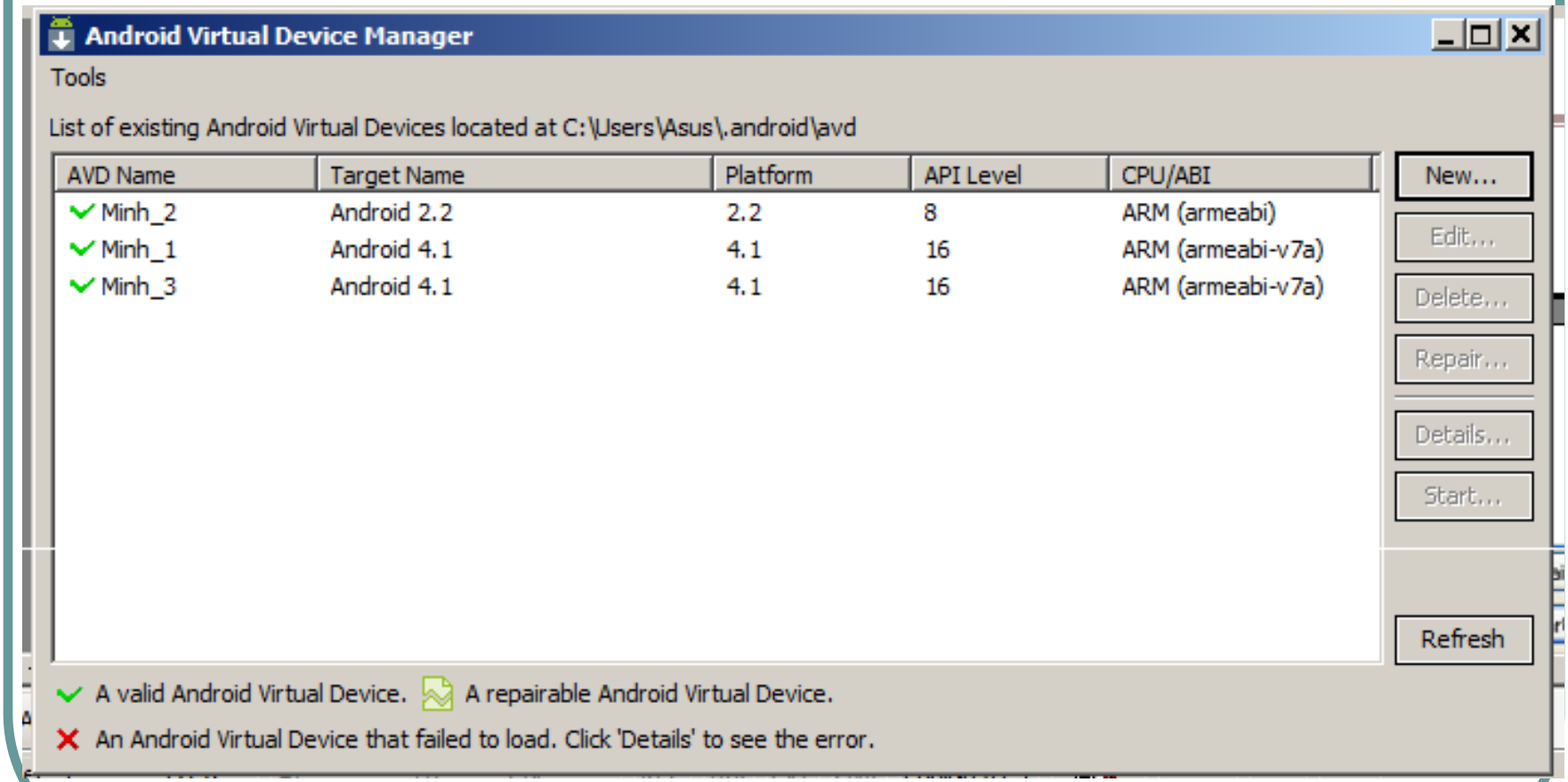
3. Tạo thiết bị giả lập

- Chạy tập tin AVD Manager.exe, trong cửa sổ xuất hiện nhấn nút New để tạo một thiết bị.
- Ở ô name ta đặt tên (chỉ được sử dụng các ký tự "a-z", "A-Z", ".-_") cho thiết bị,
- Ở hộp Target chọn phiên bản hệ điều hành, tốt nhất nên chọn phiên bản càng thấp vì như vậy khi tạo ứng dụng sẽ chạy được trên nhiều phiên bản khác nhau và tốc độ của thiết bị giả lập cũng nhanh hơn (theo thử nghiệm thì phiên bản 2.2 và 2.3 là khá tốt).
- Ở SD Card ta nhập dung lượng thẻ nhớ,
- Ở Skin chọn kích cỡ màn hình.
- Cuối cùng nhấn nút Create AVD để hoàn tất. Sau khi tạo thiết bị xong ta chọn nó rồi nhấn nút Start để chạy thiết bị.


Create new Android Virtual Device (AVD)
Name:
Minh_2
Target:
Android 2.2 - API Level 8
CPU/ABI:
ARM (armeabi)
SD Card:
Size:
4
GiB
File:
Browse...
Snapshot:
Enabled
Skin:
Built-in:
Default (WVGA800)
Resolution:
x
Hardware:
Property
Value
Abstracted LCD density
240
Max VM application heap size
24
New...
Delete


Create new Android Virtual Device (AVD)
Name:
Minh_3
Target:
Android 4.1 - API Level 16
CPU/ABI:
ARM (armeabi-v7a)
SD Card:
Size:
16
MiB
File:
Browse...
Snapshot:
Enabled
Skin:
Built-in:
HVGA
Resolution:
x
Hardware:
Property
Value
New...
Delete
Abstracted LCD density
160
Max VM application heap size
48
Device ram size
512
Override the existing AVD with the same name
Create AVD
Cancel


Android Virtual Devices Manager
Result of creating AVD 'Minh_3':
Created AVD 'Minh_3' based on Android 4.1, ARM (armeabi-v7a) processor,
with the following hardware config:
hw.lcd.density=160
vm.heapSize=48
hw.ramSize=512
OK

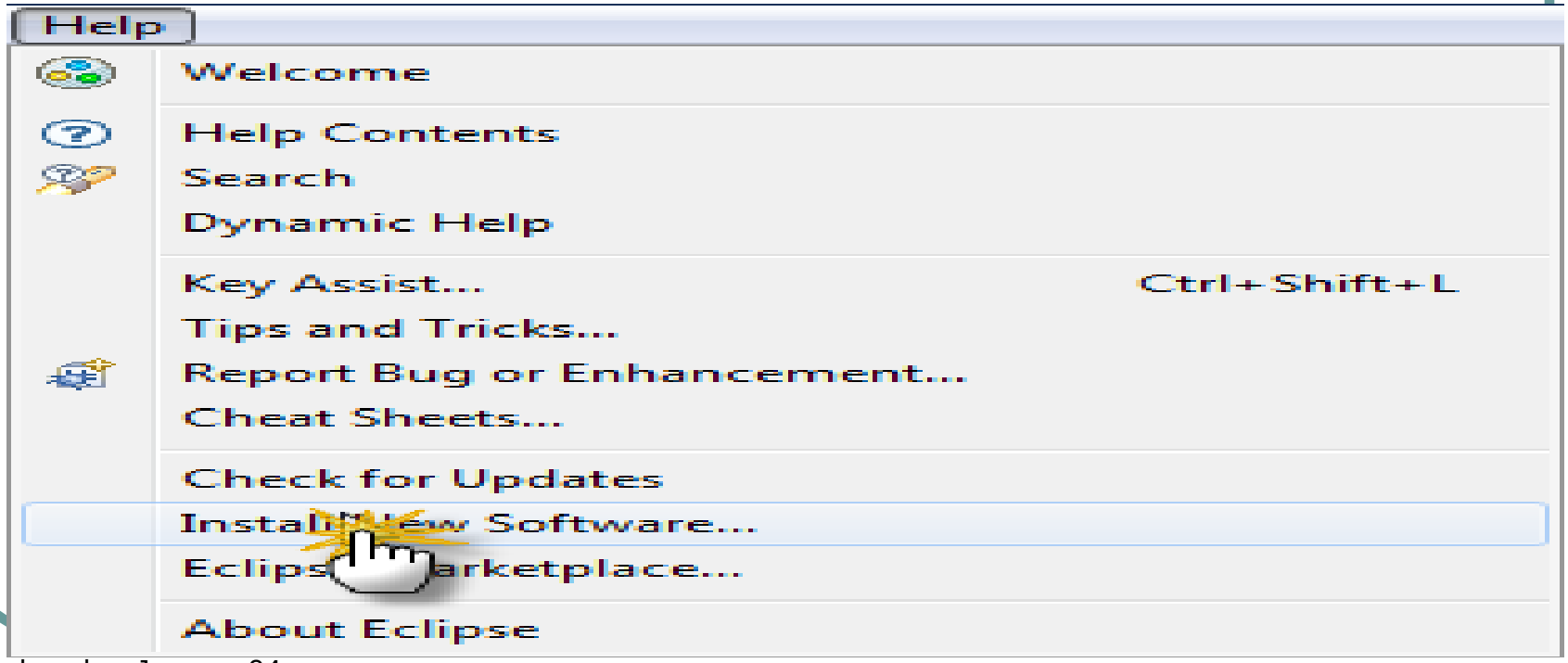

Android Virtual Device Manager
Tools
List of existing Android Virtual Devices located at C:\Users\Asus\.android\avd
AVD Name
Target Name
Platform
API Level
CPU/ABI
Minh_2
Android 2.2
2.2
8
ARM (armeabi)
Minh_1
Android 4.1
4.1
16
ARM (armeabi-v7a)
Minh_3
Android 4.1
4.1
16
ARM (armeabi-v7a)
New...
Edit...
Delete...
Repair...
Details...
Start...
Refresh
A valid Android Virtual Device.
A repairable Android Virtual Device.
An Android Virtual Device that failed to load. Click 'Details' to see the error.

4. Thêm biến môi trường

- (1) Nhấn chuột phải vào biểu tượng My Computer chọn Properties.
- (2) Trong cửa sổ System chọn Advanced system settings, ở thẻ Advanced chọn Environment Variables.
- (3) Ở cửa sổ xuất hiện, trong mục **User variables** nhấn nút **New**, → nhập **JAVA_Path** vào ô **Variable name** → nhập đường dẫn tới thư mục chứa project của JDK vào ô **Variable value**, (ví dụ **.;D:\Users\JavaSE;**) → nhấn OK. → Nhấn nút New lần nữa → nhập **ANDROID_SDK_Path** vào ô **Variable name** → nhập đường dẫn tới thư mục chứa các thiết bị giả lập đã được tạo vào ô **Variable value** (thông thường là **C:\Users\<user name>\.android\avd;**).
- (4) Tại mục **System variables,** nhấn nút **New**, → nhập **JAVA_Home** vào ô **Variable name** → nhập đường dẫn tới thư mục đã cài đặt JDK vào ô **Variable value**, (**android\avd;C:\Program Files\Java\jdk1.7.0_01\bin;** ;) → nhấn OK. → nhấn nút New lần nữa → nhập **ANDROID_SDK_HOME** vào ô **Variable name** → nhập đường dẫn tới thư mục chứa Android SDK đã được tạo vào **ô Variable value** (thông thường là **C:\Program Files\android-sdk-windows\tools**). Nhấn OK ba lần để hoàn tất.
- **Lưu ý:** Các đường dẫn trên chỉ là ví dụ minh họa, có thể thay đổi trên các máy tính khác.

5. Cài Đặt ATD plugin

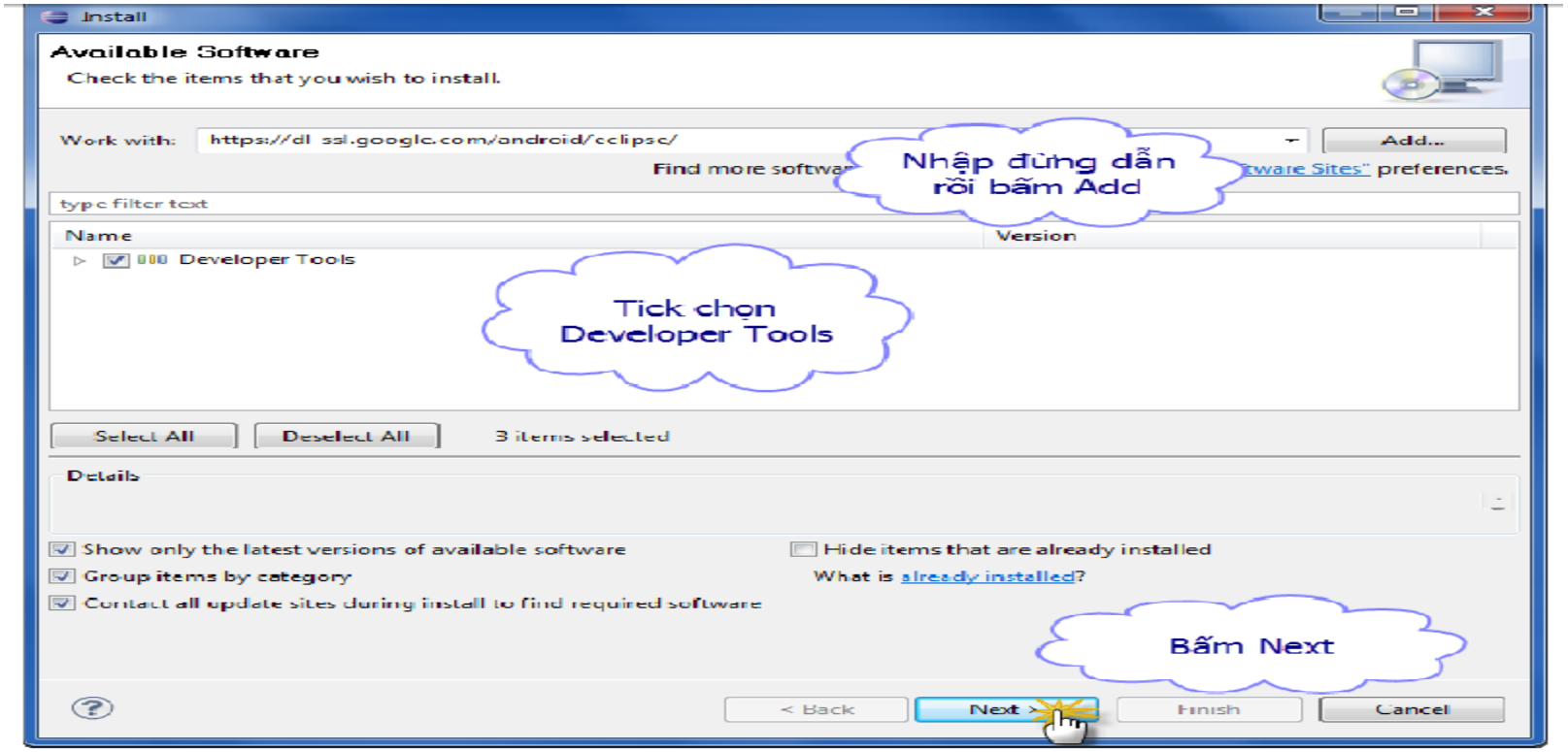
- Cài đặt Plugin cho eclipse.
- Từ giao diện eclipse ta vào menu Help và chọn Install new software.

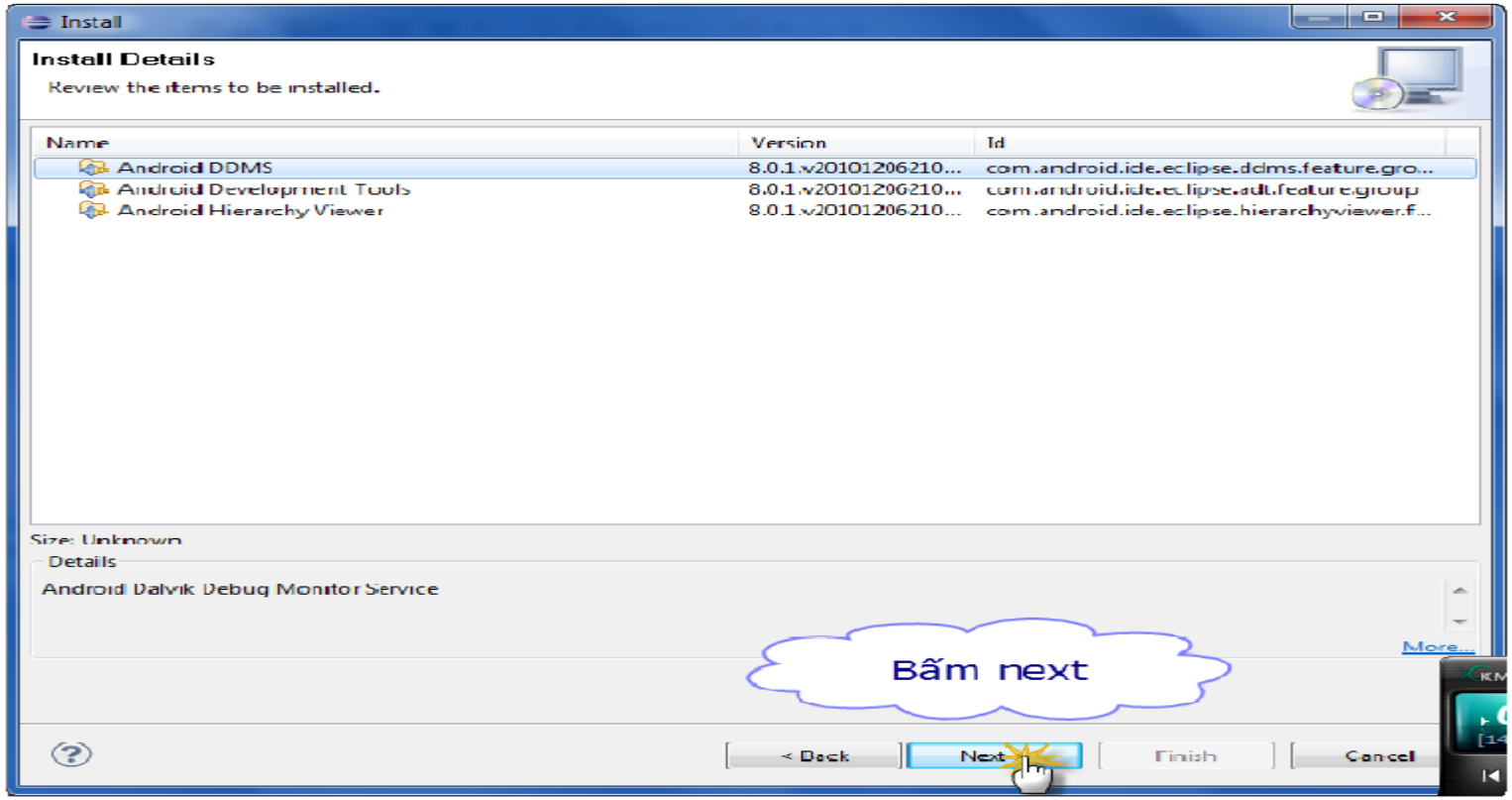


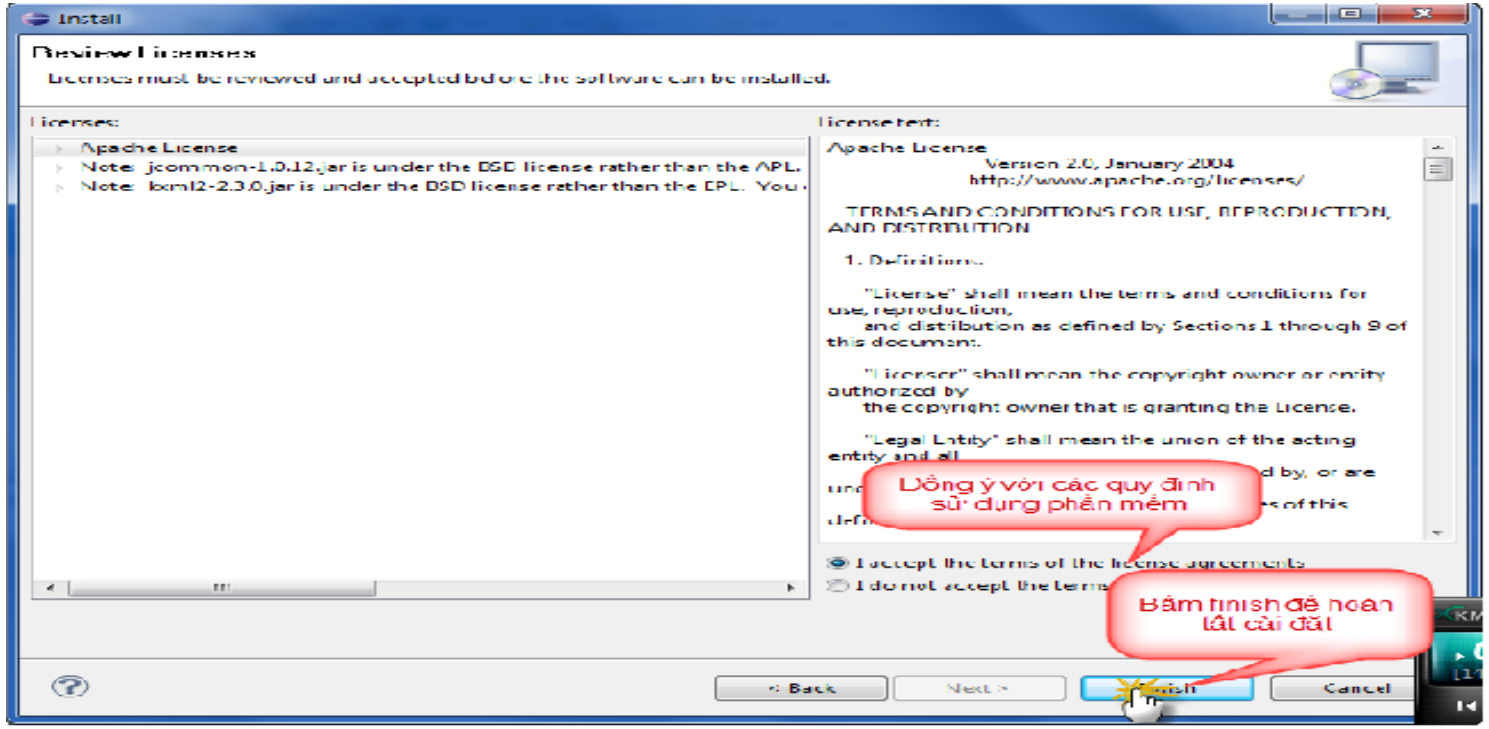
5. Cài Đặt ATD plugin(2)

- Trong cửa sổ Install ta nhập đường dẫn

  https://dl-ssl.google.com/android/eclipse/ vào ô Location with, click chọn Developer Tools rồi bấm Next


Install
Available Software
Check the items that you wish to install.
Work with:
https://dl-ssl.google.com/android/eclipse/
Add...
Find more softwa
tware Sites" preferences.
Nhập đừng dẫn rồi bấm Add
type filter text
Name
Version
Developer Tools
Tick chọn Developer Tools
Select All
Deselect All
3 items selected
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Contact all update sites during install to find required software
Bấm Next
< Back
Next >
Finish
Cancel


Install
Install Details
Review the items to be installed.
Name
Version
Id
Android DDMS
8.0.1.v20101206210...
com.android.ide.eclipse.ddms.feature.gro...
Android Development Tools
8.0.1.v20101206210...
com.android.ide.eclipse.adt.feature.group
Android Hierarchy Viewer
8.0.1.v20101206210...
com.android.ide.eclipse.hierarchyviewer.f...
Size: Unknown
Details
Android Dalvik Debug Monitor Service
More...
Bấm next
< Back
Next >
Finish
Cancel


Install
Review Licenses
Licenses must be reviewed and accepted before the software can be installed.
Licenses:
Apache License
Note: jcommon-1.0.12.jar is under the DSD license rather than the APL.
Note: lxml2-2.3.0.jar is under the DSD license rather than the EPL. You
License text:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all
Đồng ý với các quy định sử dụng phần mềm
I accept the terms of the license agreements
I do not accept the terms
Bấm finish để hoàn tất cài đặt
< Back
Next >
Finish
Cancel

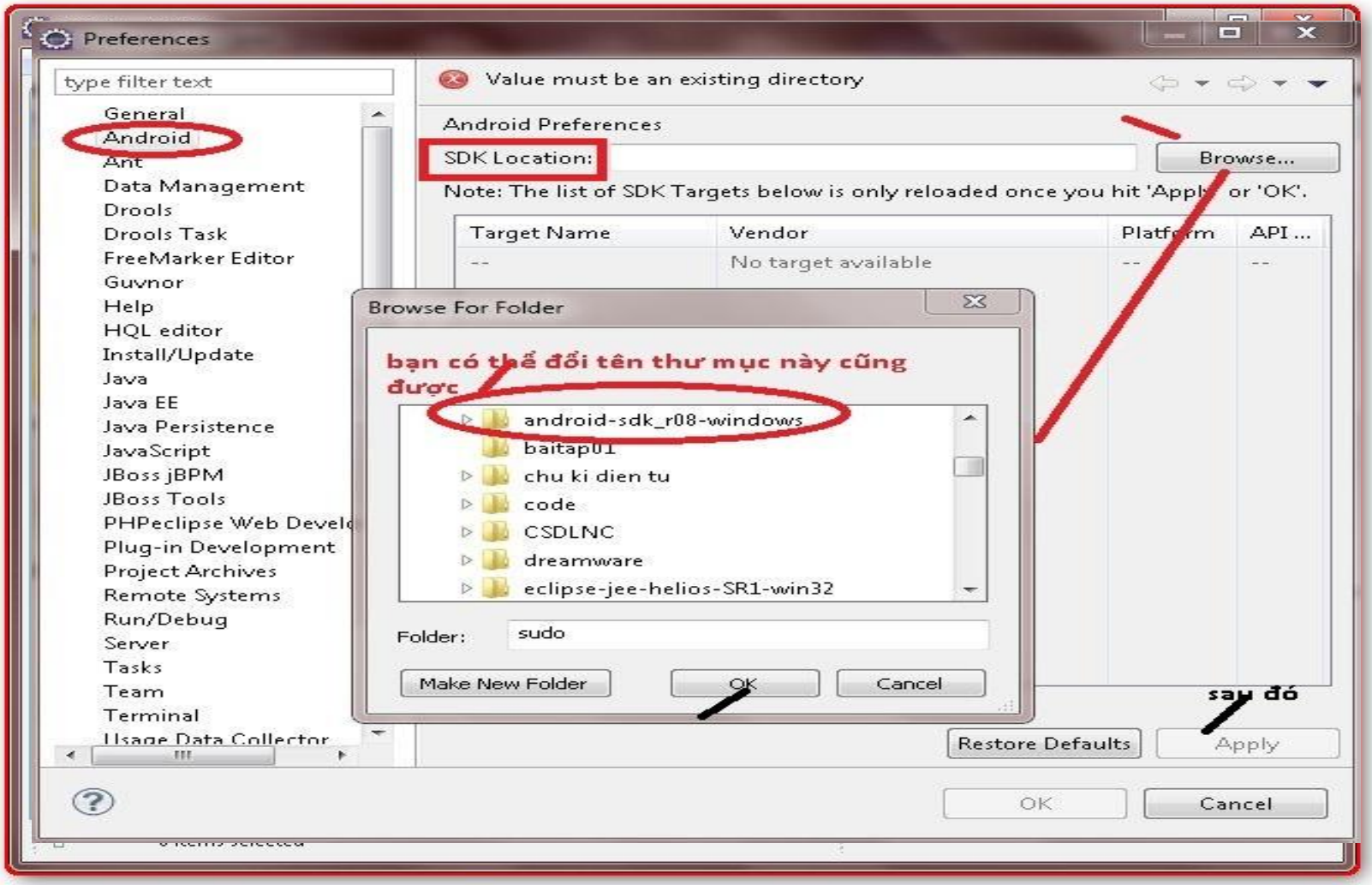
6. Cài đặt Android SDK từ Eclipse

- Tải Android SDK tại link này http://developer.android.com/sdk/index.html (nhớ chú ý là nó chia ra Window, Mac và Linux nha-bạn chọn windows).

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

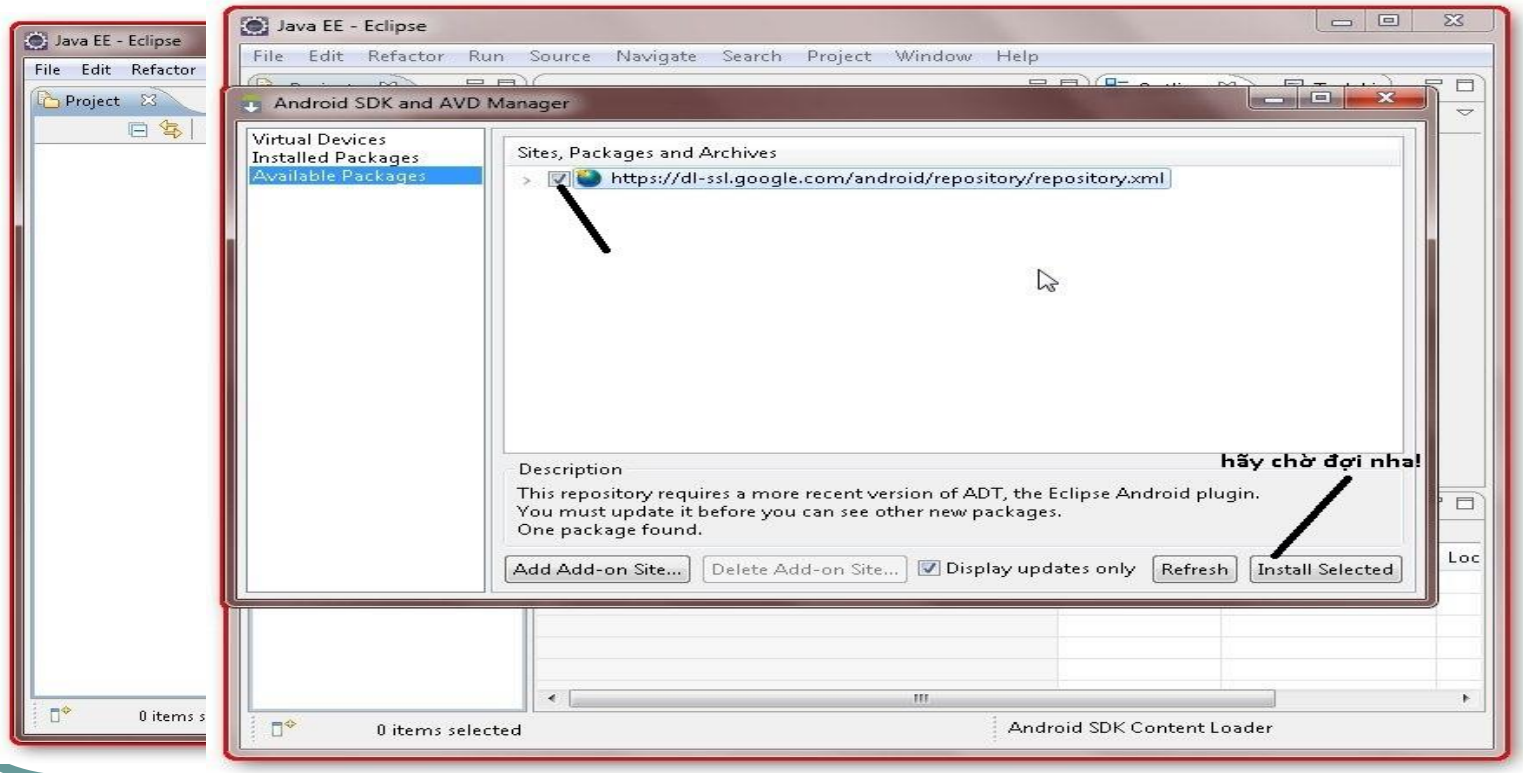
- Mở Eclipse → window → preferences → Android → trong mục SDK location chọn Browse → đưa đường dẫn đến thư mục SDK(thư mục mà bạn mới down về và giải nén đó) → bấm apply.

6. Cài đặt Android SDK từ Eclipse(2)



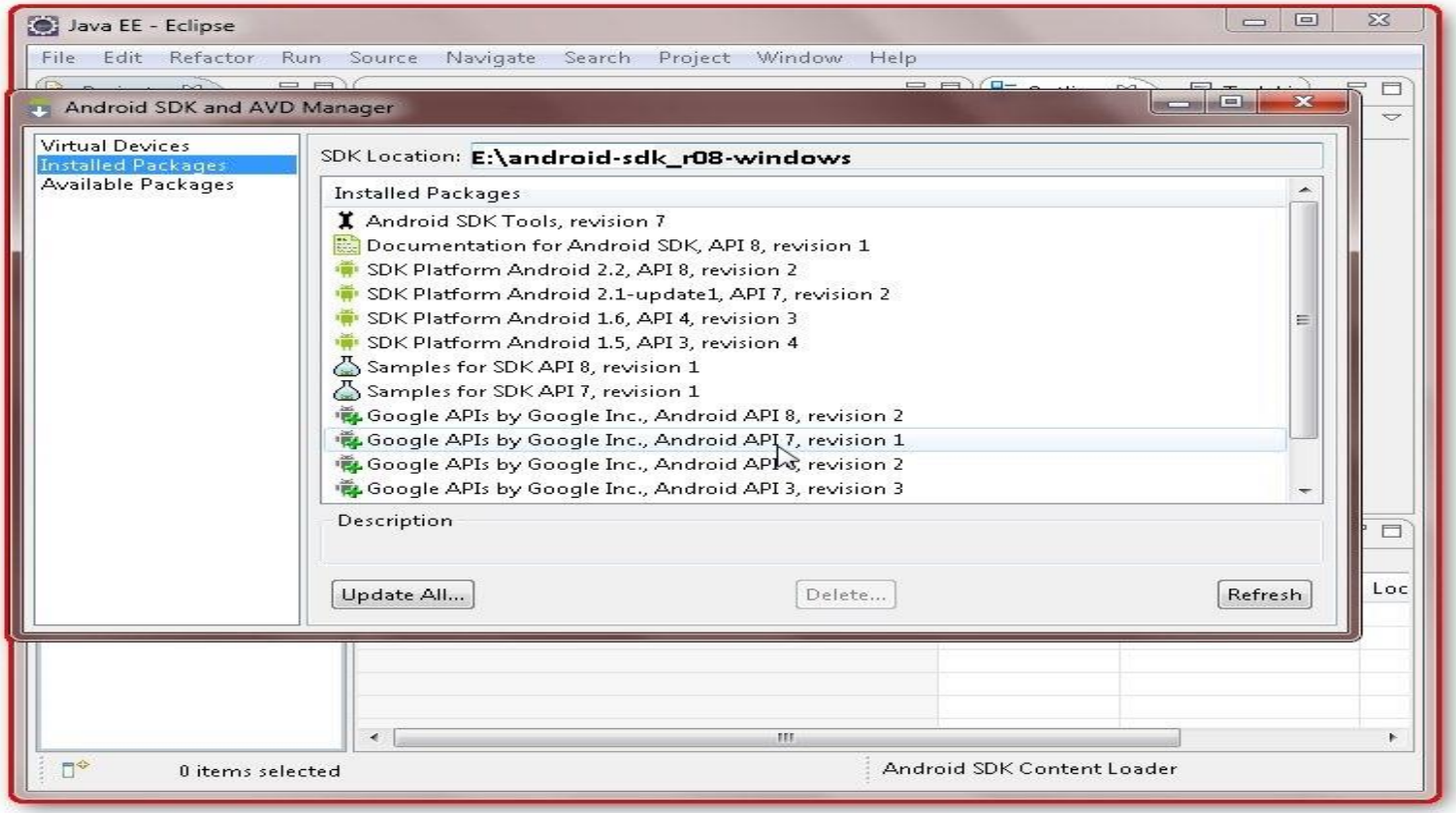
6. Cài đặt Android SDK từ Eclipse (3)

- Eclipse → window → Android SDK and AVD manager → Available packages → chọn hết và cài hết (hơi lâu một chút).



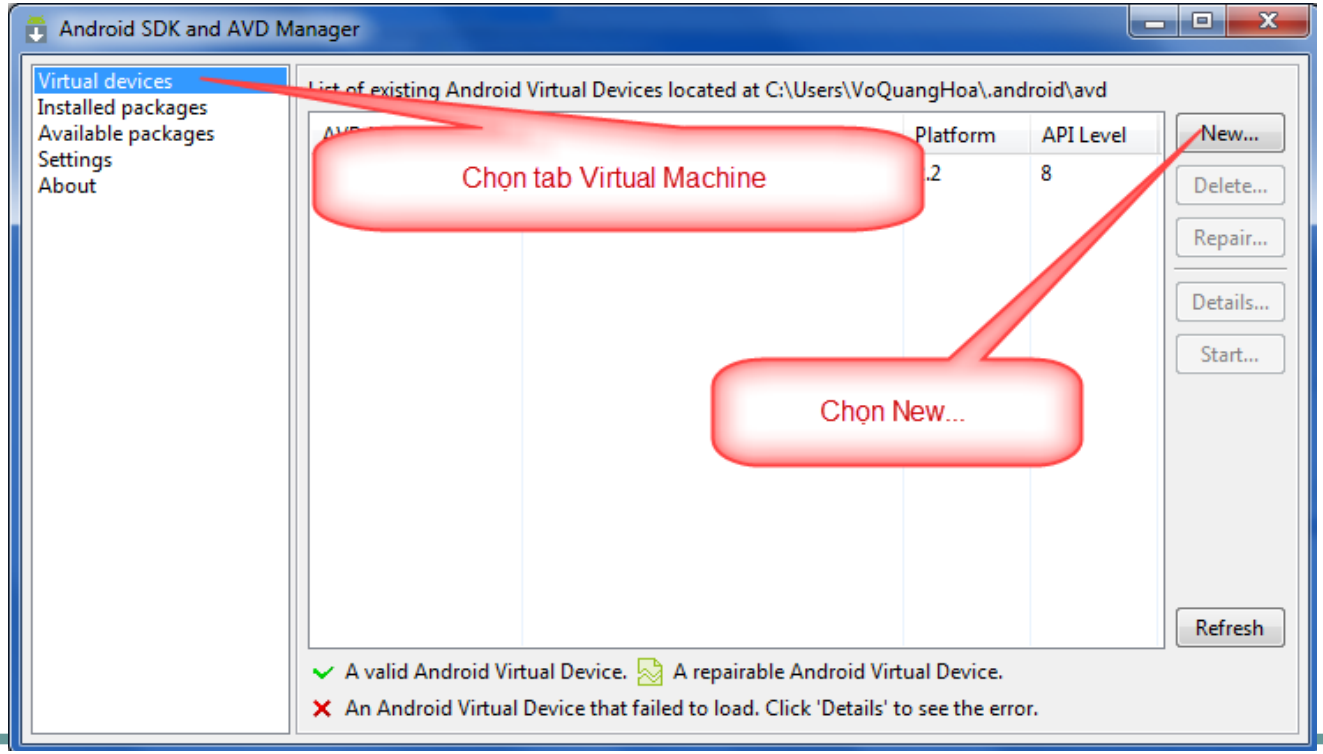
6. Cài đặt Android SDK từ Eclipse (4)

Sau khi xong thì bấm vào Installed packages sẽ thấy như sau:



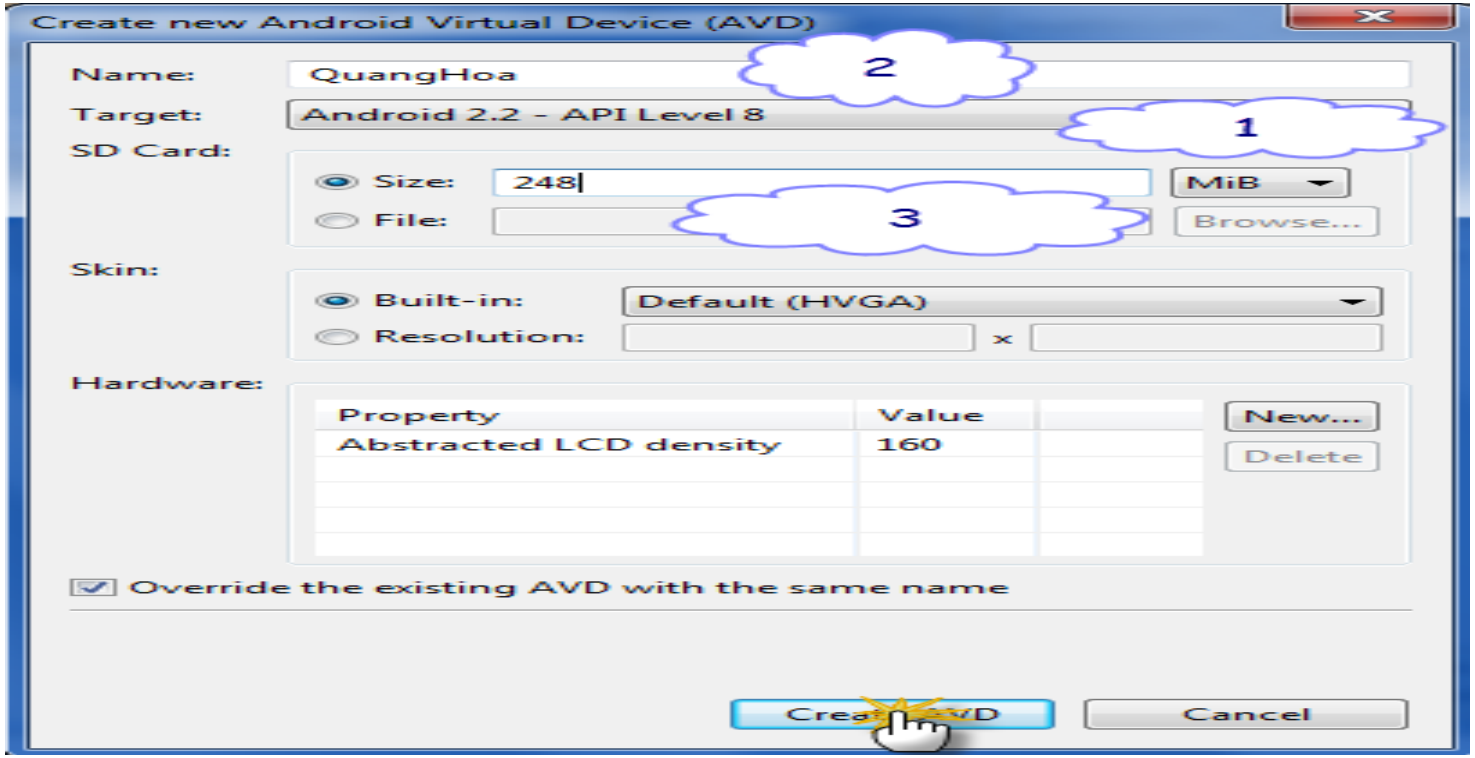
7. Tạo một điện thoại ảo(AVD) từ Eclipse

- Eclipse → window → Android SDK and AVD manager → ở cái cửa sổ mới đó, chọn mục Virtual devices → bấm nút New…



7. Tạo một điện thoại ảo(AVD) từ Eclipse (2)

- Trong hộp thoại Create new Android Virtual Machine ta nhập các thông tin theo thứ tự như hình:



7. Tạo một điện thoại ảo(AVD) từ Eclipse (3)

- Target : Nền tảng đã cài đặt, ta chọn Android 2.2 API Level 8 là nền mà ta đã cài đặt trước đó.
- Name : Là tên của máy ảo cần tạo
- SD Card : Dung lượng tối đa của thẻ nhớ ảo. (Thực chất khi chạy trên máy ảo, thẻ nhớ được sử dụng dưới dạng một file iso).
- Skin : Chọn giao diện (Phần này không quan trọng, để mặc định)
- Hardware : Chọn cấu hình điện thoại. Tại thời điểm hiện tại thì có thể để mặc định.
- Xong tất cả, bấm Create AVD

5. Hướng dẫn tạo Project Android

- Đầu tiên vào Eclipse->File > New > Project>Android>Android Project gồm có:

- Project name: đặt tên tùy bạn.

- Build Target: bạn chọn phiên bản android

- API tùy chọn.

- Application name: tên ứng dụng tùy chọn

- Package name: tên gói ban đầu(bạn phải đặt tên 2 hoặc 3 gói - chồng lên nhau. Vd: com.google hoặc sudo.root.kaka

- Create Activity: lớp ban đầu trong gói trên

- Min SDK Version: không cần điền vào cũng được,Hoặc điền vào thì chỉ rõ API bao nhiêu .ví dụ Android 2.2->API:8

- Tham khảo ví dụ Hello World: http://developer.android.com/resources/tutorials/hello-world.html

Phần III: Demo

QUESTION?