

# M9 SDK 使用说明

M9 SDK 使用说明.....	1
安装 M9 SDK.....	2
系统需求.....	2
M9SDK 安装.....	2
M9 模拟器.....	3
创建 M9 模拟器.....	3
运行 M9 模拟器.....	4
安装 M9 设备驱动.....	6
M9SDK 开发指引.....	11
应用程序添加高清图片资源.....	11
Meizu 定制类.....	11
文件管理器调用接口.....	11

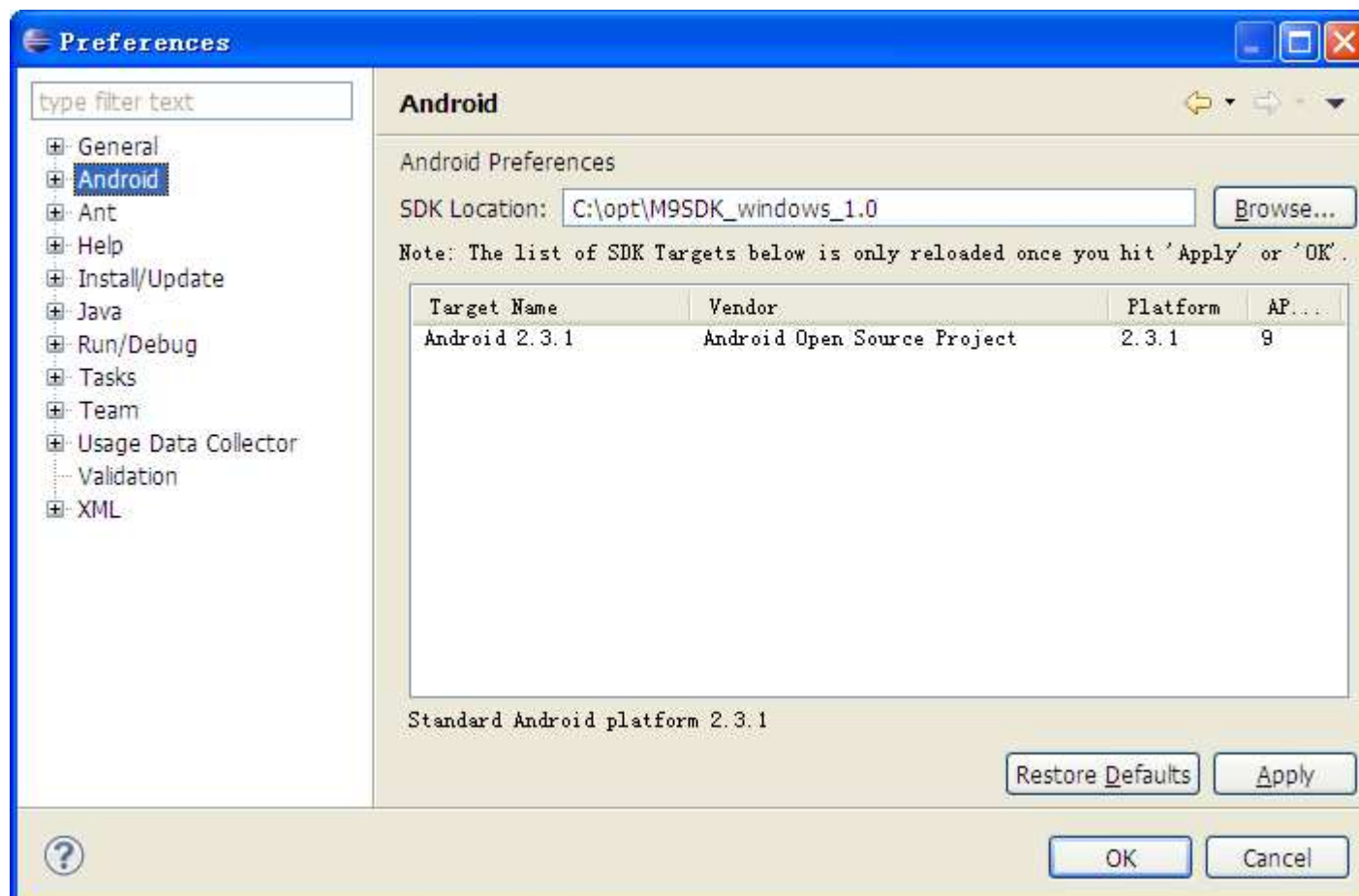
## 安装 M9 SDK

### 系统需求

1. 操作系统
  - Windows XP (32 位), Vista (32 或 64 位), Windows 7 (32 或 64 位)
  - Linux (在 Ubuntu Lucid Lynx 测试)
2. 开发环境
  - JDK 6 , <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
  - Eclipse IDE for Java Developers , <http://www.eclipse.org/downloads/>
  - Android Development Tools 插件
3. 安装 Android Development Tools 插件
  - 启动 Eclipse 然后选择 **Help > Install New Software...**
  - 点击右上角的 **Add** 按钮
  - 在显示出来的对话框中, **Name** 项输入 “ADT 插件”, **Location** 项中输入以下地址: <http://dl-ssl.google.com/android/eclipse/>
  - 选中 **Developer Tools** 然后点击 **Next**
  - 在下一个窗口中你将会看到将要下载的工具列表, 点击 **Next**
  - 选中对话框下方的 **I accept the terms of the license agreement** 项接受许可协议, 点击 **Finish**
  - 当安装完成后, 重启 Eclipse

### M9SDK 安装

1. 下载 M9 SDK , <http://developer.meizu.com/> , 解压下载到的安装包到任意目录, 如 D:\opt\
2. 启动 Eclipse , 选择 **Window > Preferences** , 点击对话框左侧的 **Android** 项
3. 点击对话框右上角的 **Browse...** 按钮, 选择 SDK 的解压路径, 如 C:\opt\M9SDK\_windows\_1.0

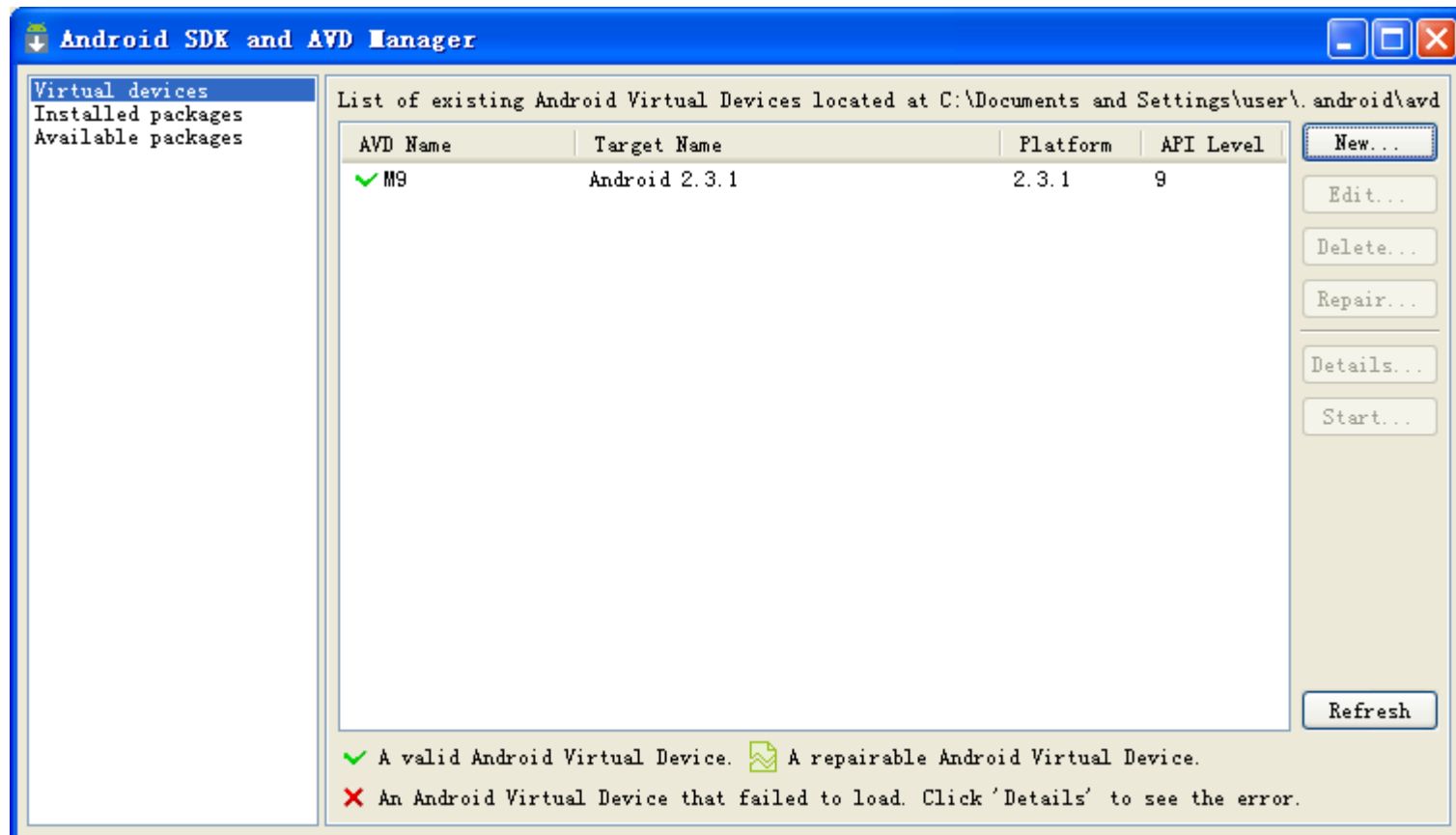


4. 点击 **OK** 结束对话框

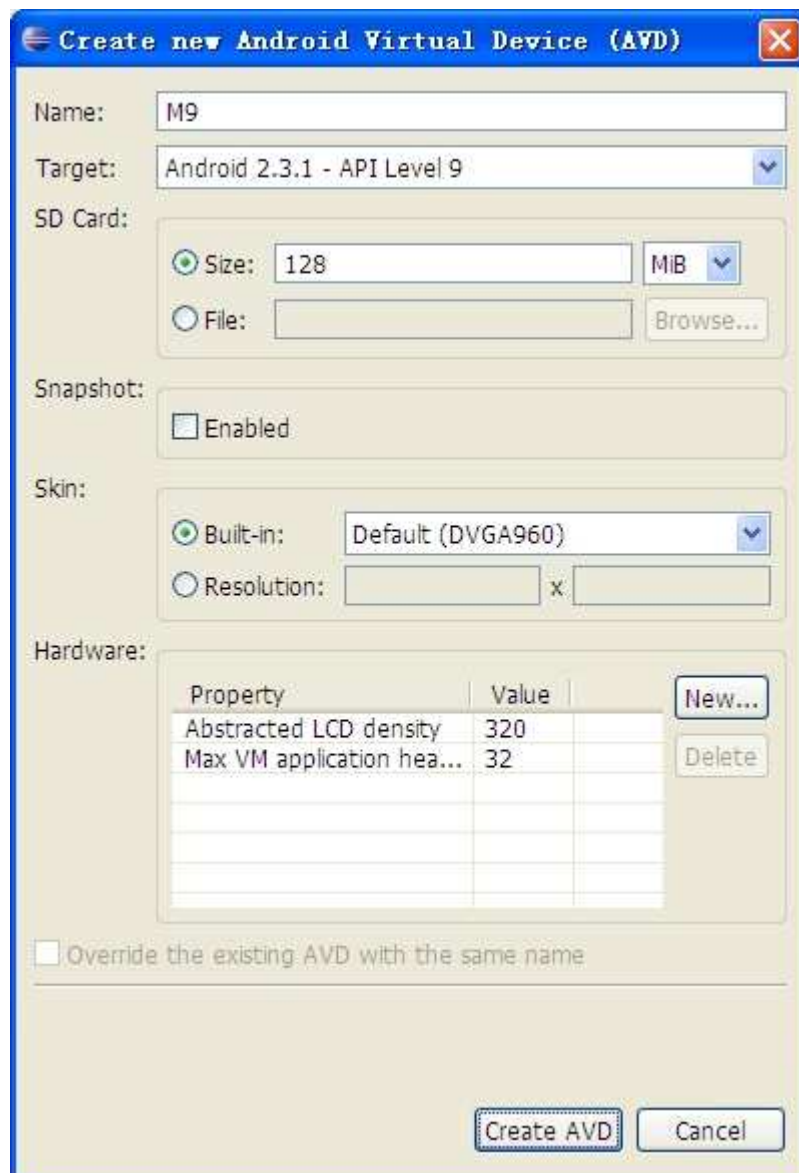
## M9 模拟器

### 创建 M9 模拟器

1. 启动 Eclipse，选择 **Window > Android SDK and AVD Manager**，点击对话框右上角的 **New...** 按钮



2. 在 Create new AVD 对话框中配置创建属性



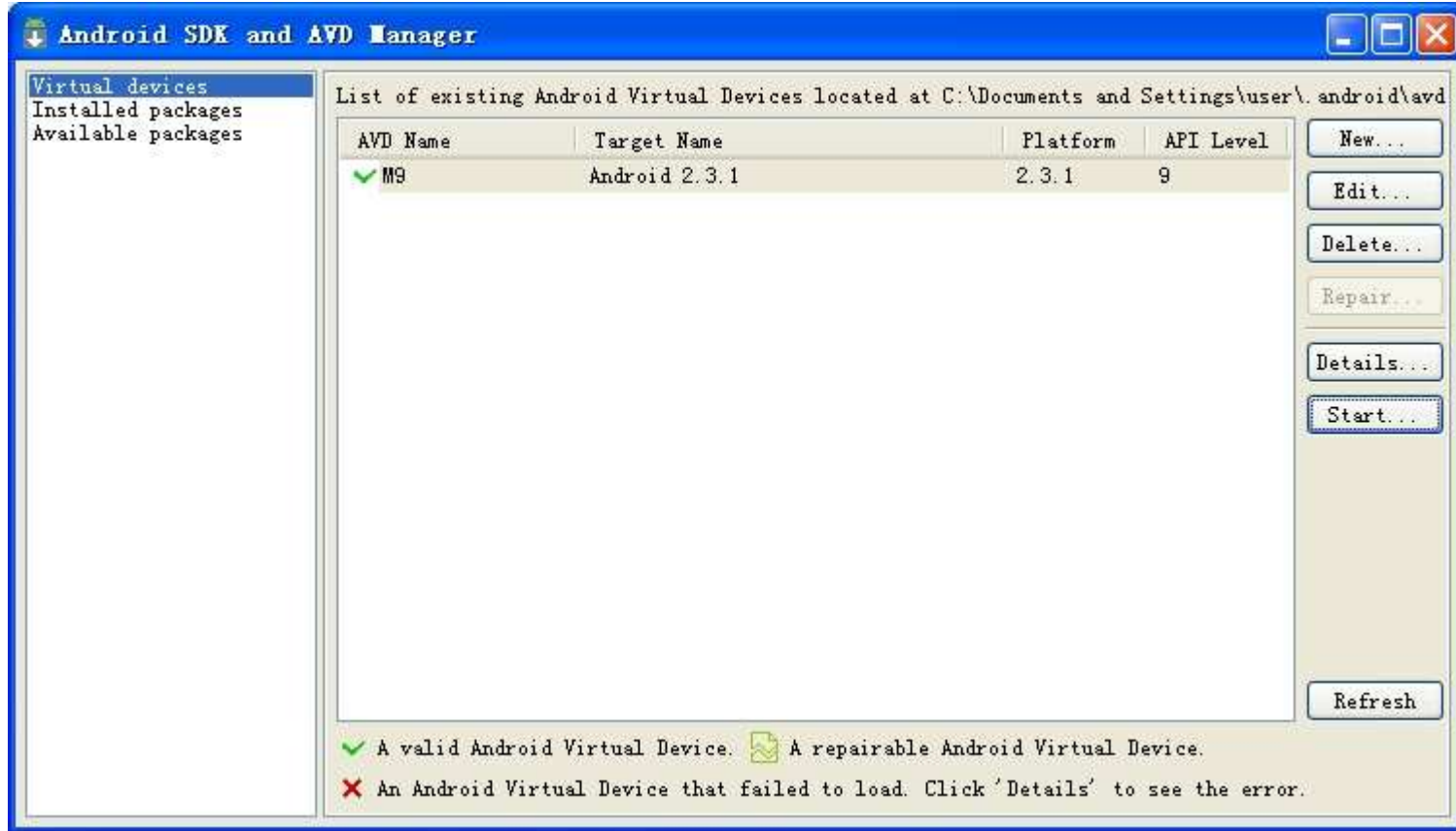
3. 点击下方的 **Create AVD** 按钮，创建模拟器

4. 在模拟器配置文件 config.ini 中增加：`hw.ramSize=512`，可以让模拟器跟 M9 设备拥有相同的运行内存容量，保证模拟器顺序启动运行。

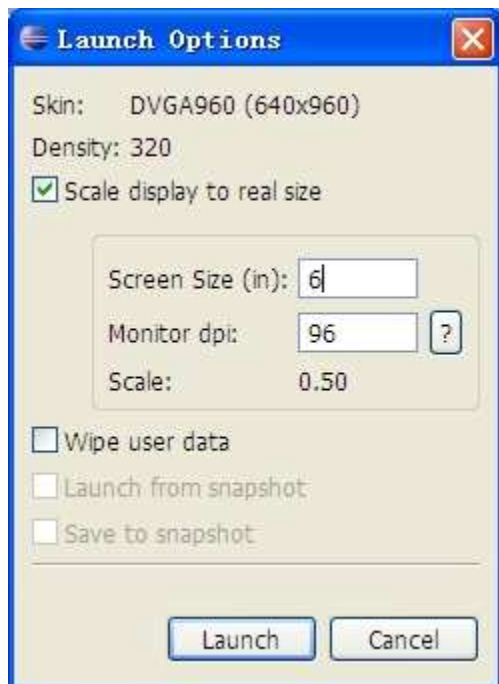
- (Windows) %USERPROFILE%\android\avd\M9.avd\config.ini
- (Linux) \$HOME/.android/avd/M9.avd/config.ini

## 运行 M9 模拟器

1. 启动 Eclipse，选择 **Window > Android SDK and AVD Manager**，选中对话框中已创建的 M9 模拟器，点击右侧的 **Start** 按钮



2. 在新对话框中选中 **Scale display to real size**，修改 **Screen Size** 为合适的大小，由于 M9 屏幕分辨率很高，所以模拟器如果点对点的话，一般的电脑屏幕显示不完整，因此需要在启动模拟器时，设置屏幕尺寸到 3.5 寸或更大一些，只要设置到适合您的电脑屏幕的尺寸即可



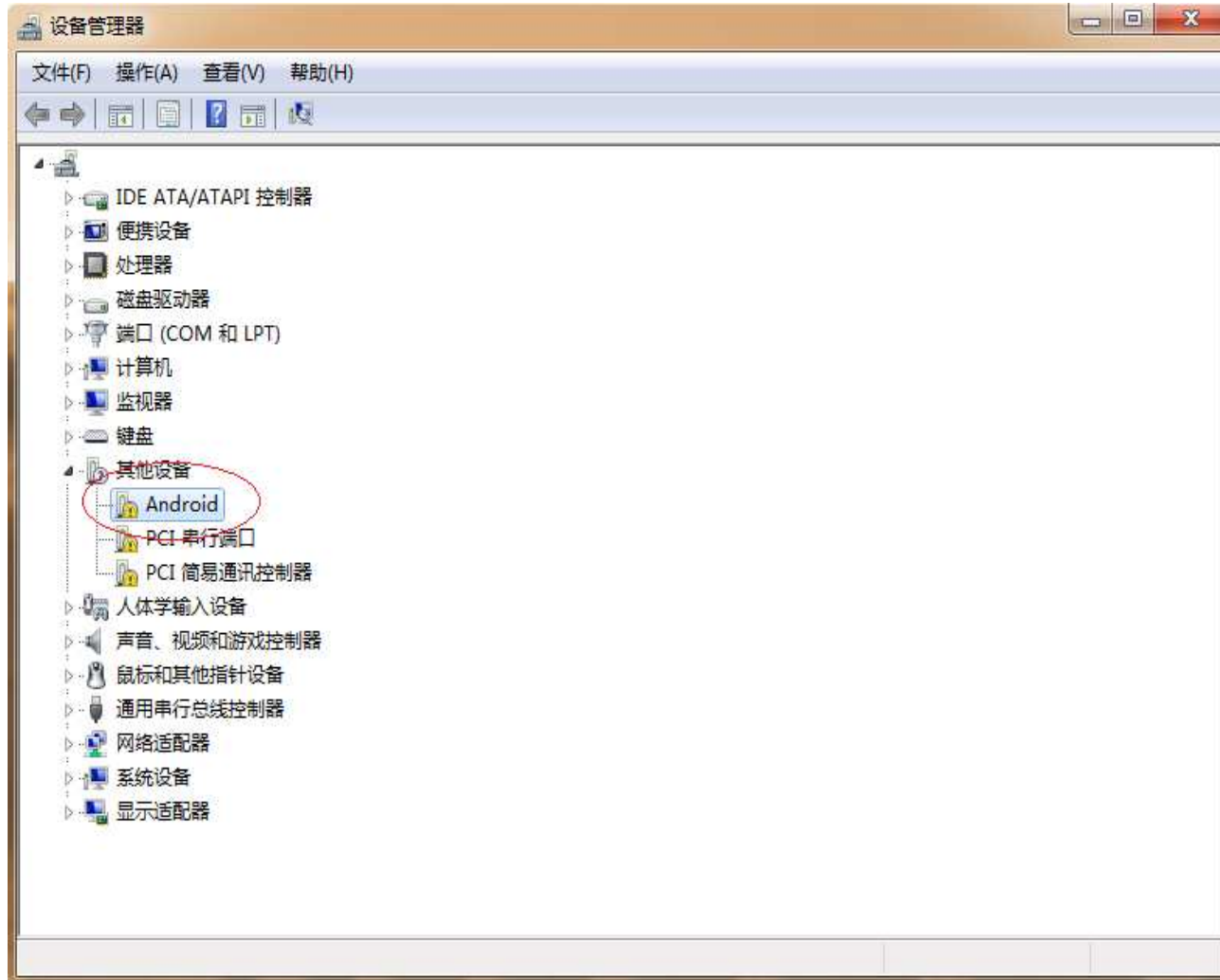
3. 点击 **Launch** 启动模拟器



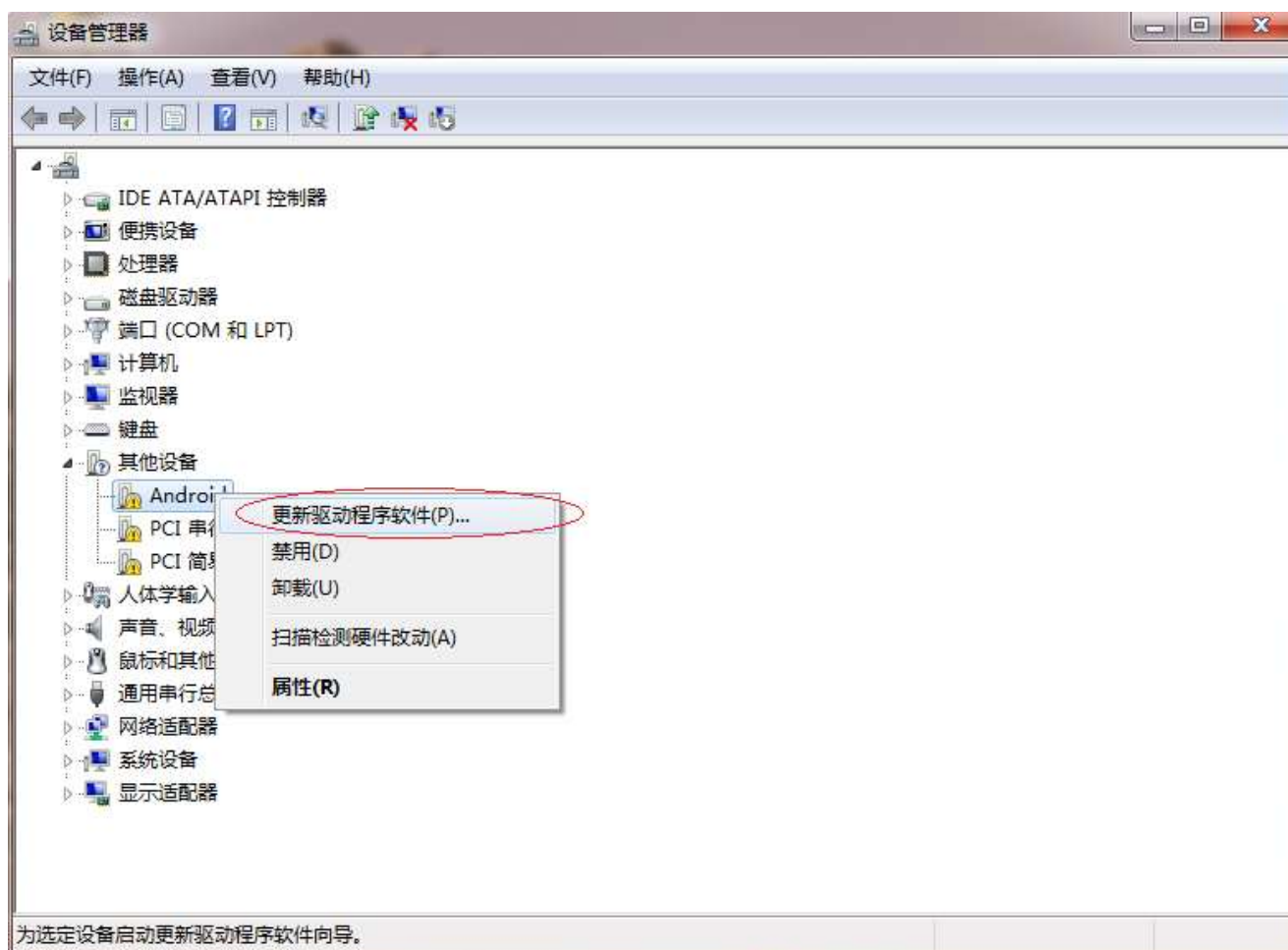
## 安装 M9 设备驱动

Windows 连接 M9 机器调试需要安装驱动，Linux 不需要。

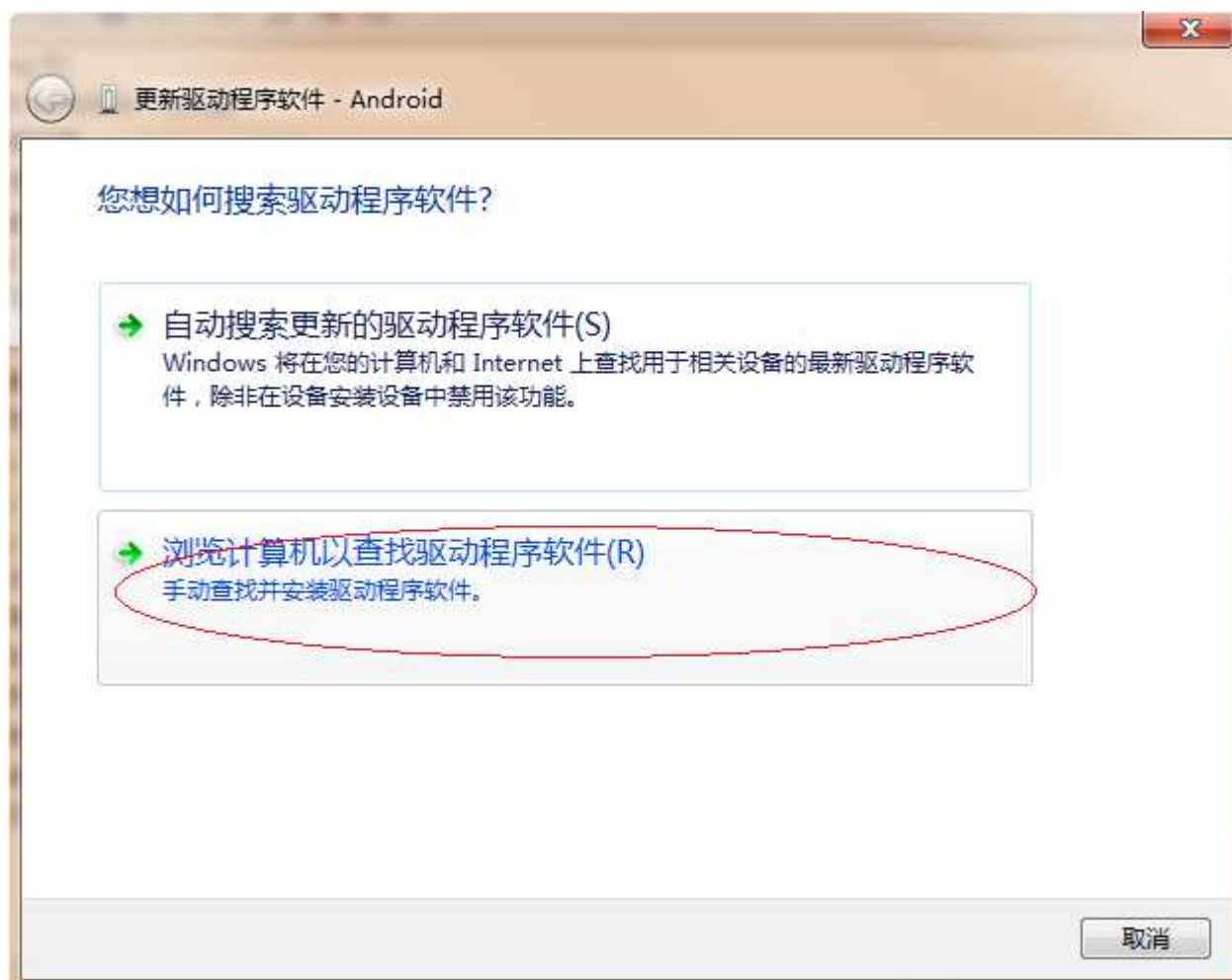
1. 开启 M9 的 USB 调试模式，设置>USB 模式>USB 调试
2. 通过 USB 连接 M9 到 PC
3. 在控制面板中打开设备管理器，在设备管理器列表中可见到有未知的 Android 设备



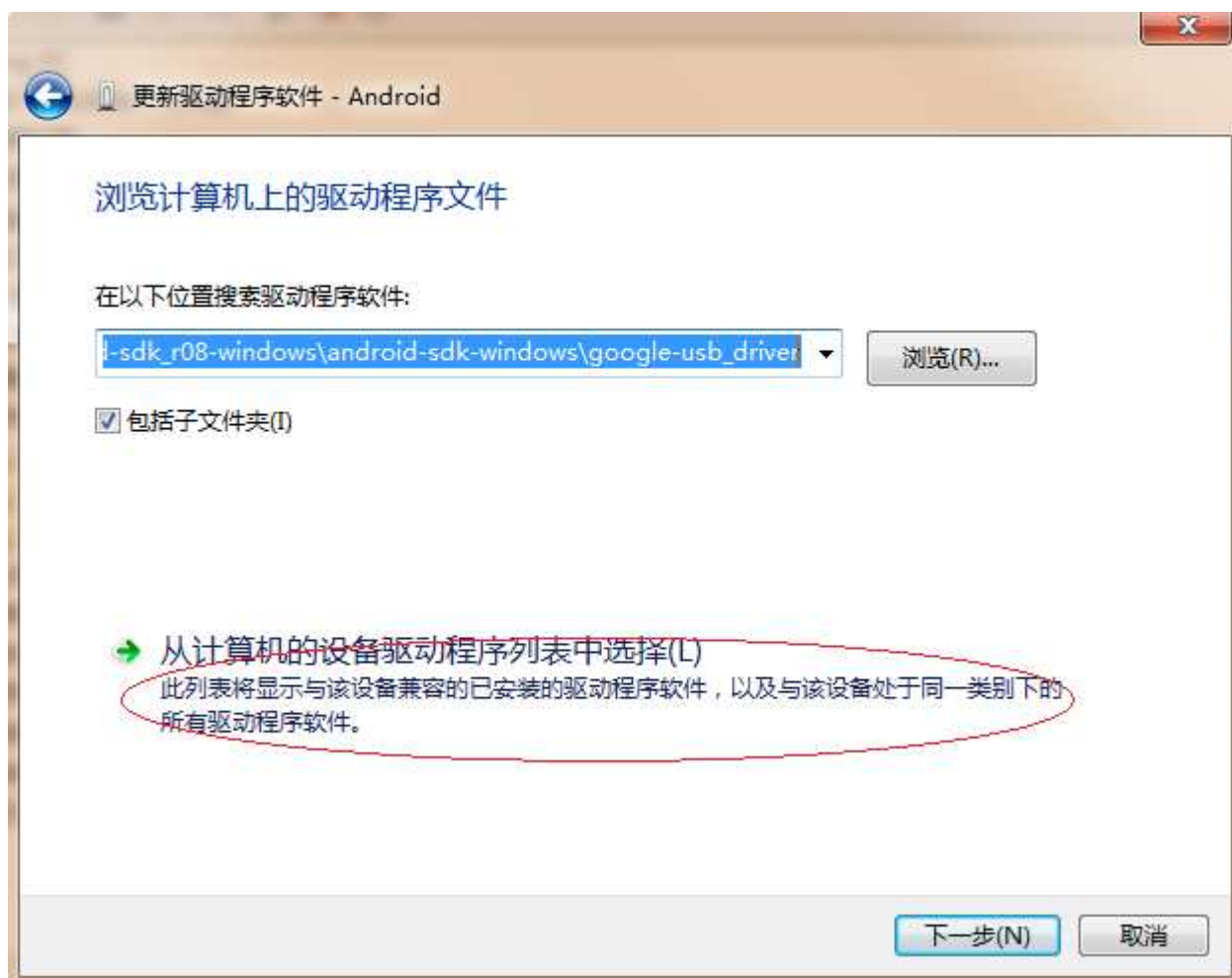
4. 在对应的设备处点击鼠标右键，选择更新驱动程序软件



5. 选择浏览计算机以查找驱动程序软件



6. 点击从计算机的设备驱动程序列表中选择



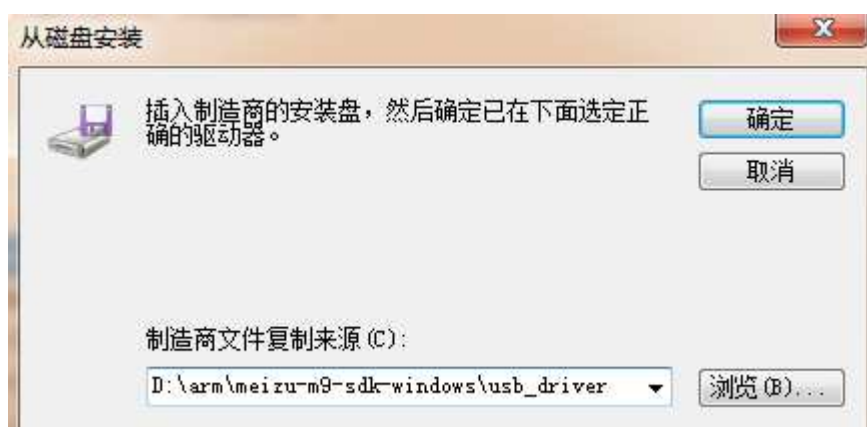
7. 点击下一步



8. 点击从磁盘安装



9. 点击浏览，选择 SDK 解压缩文件中的“usb\_driver”文件夹，点击确定





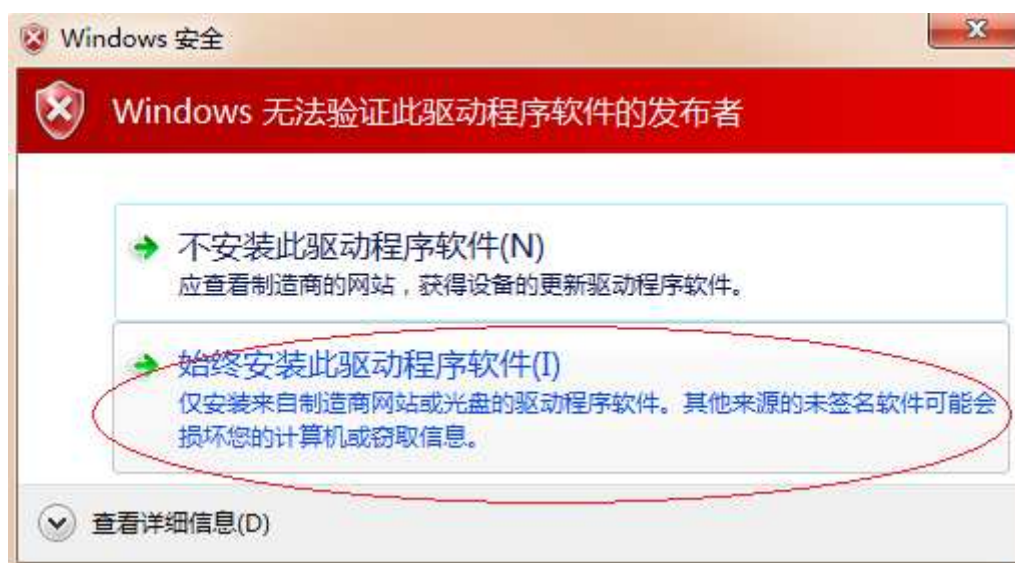
10. 点击下一步



11. 此时, 会弹出安全警告, 选择是。



12. windows 安全警告信息中选择始终安装此驱动程序软件



13. 安装成功，点击关闭



## M9SDK 开发指引

### 应用程序添加高清图片资源

由于 M9 的分辨率高达 960x640，像素密度达到了 320dpi (xhdpi)，因此必须使用特定配置才能满足高清无缩放的界面效果。而 GOOGLE 官方 SDK 模拟器并不支持显示到 320dpi 清晰度(只支持到 240dpi 即 hdpi)，因此必须使用 M9SDK 的模拟器才能最直观的开发出 M9 适用的高清版本应用程序。当然，使用 GOOGLE 官方 SDK 开发的 hdpi 清晰度的应用程序也能在 M9 上运行良好。但是如果发挥 M9 的高清屏幕的优势，则需要在应用程序中添加相应的 xhdpi 的资源。

高清资源的添加方法：

1. 在应用程序工程根目录下的 res 文件夹中新建 drawable-xhdpi 文件夹
2. 将应用所属的图片资源文件拷贝到 drawable-xhdpi 目录下
3. 重新编译工程生成 APK 文件

### Meizu 自定制类

M9SDK 包含了如下自定制类：

1. Com.meizu.app 包：TabActivityPlus
2. Com.meizu.widget 包：AlphabetIndexerPlus、ButtonBarLinearLayout、EditTextEx、SelectBarLineLayout、TabHostPlus、TabWidgetPlus、TextInfoProgressBar

### 文件管理器调用接口

文档开放了 3 个接口供外部使用：

1. 保存文件

使用说明：选择文件要保存的路径

输入参数：intent

- 1) 设置 action 为 Intent.ACTION\_MZ\_SELECT\_SAVE\_PATH ("com.meizu.action.SAVE\_FILE")
- 2) 提供了 2 个 extra 参数，用来显示居中的标题和进文档的初始路径

居中标题：Intent.EXTRA\_TITLE，不设置时带有图片和标题文字提示信息的一栏不显示

初始化路径：Intent.EXTRA\_MZ\_INIT\_PATH("init\_directory")，不设置时默认进 sdcard 目录

返回参数：

返回选择保存的路径（文件）Uri，解析示例：String dir = data.getData().getPath();

调用示例代码：

```
Intent intent = new Intent();
intent.setAction("com.meizu.action.SAVE_FILE");
intent.addCategory("android.intent.category.DEFAULT");
intent.putExtra(Intent.EXTRA_TITLE, "test save list");
intent.putExtra("init_directory", currentDirectory);
try {
    startActivityForResult(intent, CODE_SAVE_SELECT);
} catch (android.content.ActivityNotFoundException ex) {
}
```

## 2. 选择多个文件或文件夹

使用说明：进入文档选择多个文件或文件夹（可以混合选择）

输入参数：intent

1) 设置 action 为 Intent.ACTION\_MZ\_SELECT\_MULTI\_FILE ("com.meizu.action.CHOOSE\_MULTI\_FILE")

2) 提供了 2 个 extra 和 1 个 extra list 参数，用来显示居中的标题、初始路径、文件过滤信息

居中标题：Intent.EXTRA\_TITLE，不设置时带有图片和标题文字提示信息的一栏不显示

初始化路径：Intent.EXTRA\_MZ\_INIT\_PATH("init\_directory")，不设置时默认进 sdcard 目录

文件后缀过滤设置：Intent.EXTRA\_MZ\_DOC\_FILE\_FILTER ("file\_filter")，不设置时显示所有的文件和文件夹；设置后只显示文件夹和后缀名在后缀列表中包含的文件

另外，底部右边按钮默认为“打开”，如果需要修改，也可以使用 extra 参数："botton\_text"来改变。例如：intent.putExtra("botton\_text", "播放"); 如果需要文件夹不可选，只能选择文件，可以使用 extra 参数："select\_dir"。例如：intent.putExtra("select\_dir", false);

返回参数：

返回 2 个列表：选择的多个文件、文件夹的 Uri 列表，文件与文件夹分开：

调用示例代码：

```
ArrayList<String> filterList = new ArrayList<String>();
filterList.add(".ff");
filterList.add("mp3");
filterList.add("txt");

Intent intent = new Intent();
intent.setAction("com.meizu.action.CHOOSE_MULTI_FILE");
intent.addCategory("android.intent.category.DEFAULT");
intent.putExtra(Intent.EXTRA_TITLE, "test multi choice");
intent.putExtra("init_directory", "/sdcard/");
intent.putStringArrayListExtra("file_filter", filterList);

try {
    startActivityForResult(intent, CODE_CHOICE_MULTI);
} catch (android.content.ActivityNotFoundException ex) {
}

// 处理返回值
ArrayList<Parcelable> file_list = data
    .getParcelableArrayListExtra("fileList");
ArrayList<Parcelable> dir_list = data
    .getParcelableArrayListExtra("folderList");
for (Parcelable iter : file_list) {
    Log.e(tag, ((Uri) iter).getPath());
}
for (Parcelable iter2 : dir_list) {
    Log.e(tag, ((Uri) iter2).getPath());
}
```

### 3. 选择单个文件接口

使用说明：进入文档选择 1 个文件

输入参数：intent

1) 设置 action 为 Intent.ACTION\_MZ\_SELECT\_SINGLE\_FILE ("com.meizu.action.CHOOSE\_SINGLE\_FILE")

2) 与多选同，提供了 2 个 extra 和 1 个 extra list 参数，用来显示居中的标题、初始路径、文件过滤信息。

居中标题：Intent.EXTRA\_TITLE，不设置时带有图片和标题文字提示信息的一栏不显示

初始化路径：Intent.EXTRA\_MZ\_INIT\_PATH("init\_directory")，不设置时默认进 sdcard 目录

文件后缀过滤设置：Intent.EXTRA\_MZ\_DOC\_FILE\_FILTER ("file\_filter")，不设置时显示所有的文件和文件夹；设置后只显示文件

夹和后缀名在后缀列表中包含的文件

返回参数：intent

与选择保存路径接口同，返回选择文件的路径（文件）Uri，解析示例：String dir = data.getData().getPath();

调用示例代码：

```
ArrayList<String> filterList = new ArrayList<String>();
filterList.add(".ff");
filterList.add("ee");
filterList.add("Mp3");
Intent intent = new Intent();
intent.setAction("com.meizu.action.CHOOSE_SINGLE_FILE");
// intent.putExtra(Intent.EXTRA_TITLE, "test single choice");
// intent.putExtra("init_directory", "/sdcard/");
intent.putStringArrayListExtra("file_filter", filterList);
try {
    startActivityForResult(intent, CODE_CHOICE_SINGLE);
} catch (android.content.ActivityNotFoundException ex) {
}
```