

GOOGLE 手機程式設計

序

2007 年 10 月，Google Phone 即將出現的傳聞吵得沸沸揚揚，2007 年 11 月 12 日，Google 終於公布答岸，原來、Google 並非要做手機，而是直接釋出了一個基於 Linux 的手機平台 - Android，Android 平台的核心採用了 GPL v2 的授權，應用部分則採用了 Apache Software License 授權，這意味著手機產業可能會有所改變，由於已經有 33 家廠商參與 Android 的 Open Handset Alliance 聯盟，目前市場上以 Nokia 的 Symbian 平台與微軟 Windows Mobile 平台為主的局勢很可能會被打破，Google 所採用的開放原始碼架構對廠商與開發者都有相當大的吸引力，因此、Android 可望再手機平台上異軍突起，這是所有手機軟硬體開發者都應該仔細研究的一項新興技術。

Google 手機除了採用 Linux 平台為核心以外，也使用了 Java 作為開發的主要語言（到作者截稿前、更是唯一的應用程式開發語言），而 Java 也從 2007 年初開始就已經成為開放原始碼的成員之一，從這點看來、Google 使用開放原始碼的意圖相當明顯，然而、為了避免強制開放原始碼的限制，使得手機製造商與軟體設計廠商被迫要開放原始碼，因此、在應用端的 Java 平台以上，Android 所使用的是 Apache Software License (ASL)_{[1][2]}，ASL 並不要求軟體開發者要開放原始碼，這使得軟體開發廠商可以透過開發 Google 手機程式營利。

Google 在手機開發平台上的這種佈局是相當精巧而有彈性的，一方面藉助開放原始碼社群的力量，另一方面又可以吸引手機製造公司與軟體設計公司紛紛加入戰局，而不需要受到開放原始碼與平台授權金的限制，這使得 Google Phone – Android 平台具有相當大的吸引力，相信在 2008 年中 Google Phone 開始量產販售後，我們應該可以看到手機產業的大轉變。

本書將由淺入深的講解 Google 手機平台上的程式設計原理與方法，並以範例導向的方式，引導程式設計者進入 Google Phone 的程式設計殿堂。

內容

Google 手機程式設計	1
序	1
第 1 章 Android 平台簡介	6
1.1Android 平台的結構	6
1.2Google Phone 的外觀與功能	7
1.3 啓動 Google Phone 的模擬器	10
第 2 章程式開發環境的安裝	13
2.1 安裝 Java 的 JDK	13
2.2 安裝 Android 的 SDK	20
2.3 安裝 Eclipse 整合開發環境	24
2.4 在 Eclipse 中安裝 Android 外掛元件	28
	29
第 3 章開發您的第一個程式	40
3.1 建立 Android 程式專案	40
3.2 修改 XML 介面	43
3.3 執行專案程式	45
3.4 程式執行的原理	48
第 4 章 Android 核心物件與程式架構	50
4.1Android 的系統架構	50
4.2 程式核心 - Activity 物件	50
4.3 重要的元件簡介	50
第 5 章視覺化介面的基礎	54
5.1Android 介面設計簡介	54
5.2 以 XML 設計視覺化介面	55
5.3 基本的視覺元件	55
5.4 各種排版元件 - Layout	55
5.4.1FrameLayout	56
5.4.2AbsoluteLayout	58
5.4.3LinearLayout	60
5.4.4RelativeLayout	61
5.4.5TableLayout	63
第 6 章視覺化介面進階	67

6.1 視覺元件	67
6.2 文字型元件	67
6.2.1 TextView	67
6.2.2 AutoCompleteTextView	69
6.3 圖片相關元件	71
6.4 資料綁定元件	72
6.5 XML 屬性的設定	73
6.6 統一的風格元件	73
第 7 章事件處理	74
7.1 事件處理的基本概念	74
7.2 按鈕事件	74
7.3 滑鼠事件	75
7.4 鍵盤事件	75
7.5 時間事件	80
7.6 Notification 通知事件的處理	83
第 8 章資源的使用	83
8.1 Android 中的資源概念	83
8.2 Simple Values	83
8.3 Drawables	83
8.4 Animation	83
8.5 Layout	83
8.6 Style	83
第 9 章系統程式相關物件	85
9.1 Android 的系統程式簡介	85
9.2 程式核心 - Activity 物件	85
9.3 Bundle	85
9.4 Message	85
9.5 連接者 - Intent 物件	85
9.6 服務 - Service 物件	85
9.7 資料提供者 - Provider 物件	85
9.8 AlarmManager	85
第 10 章字串處理與 XML	86
10.1 Android 中的字串相關函式庫	86
10.2 字串物件的使用	86
10.3 正規表示式	86
10.4 處理 XML 文件	86
第 11 章資料儲存	88
11.1 共用資料 - SharedPreference	88

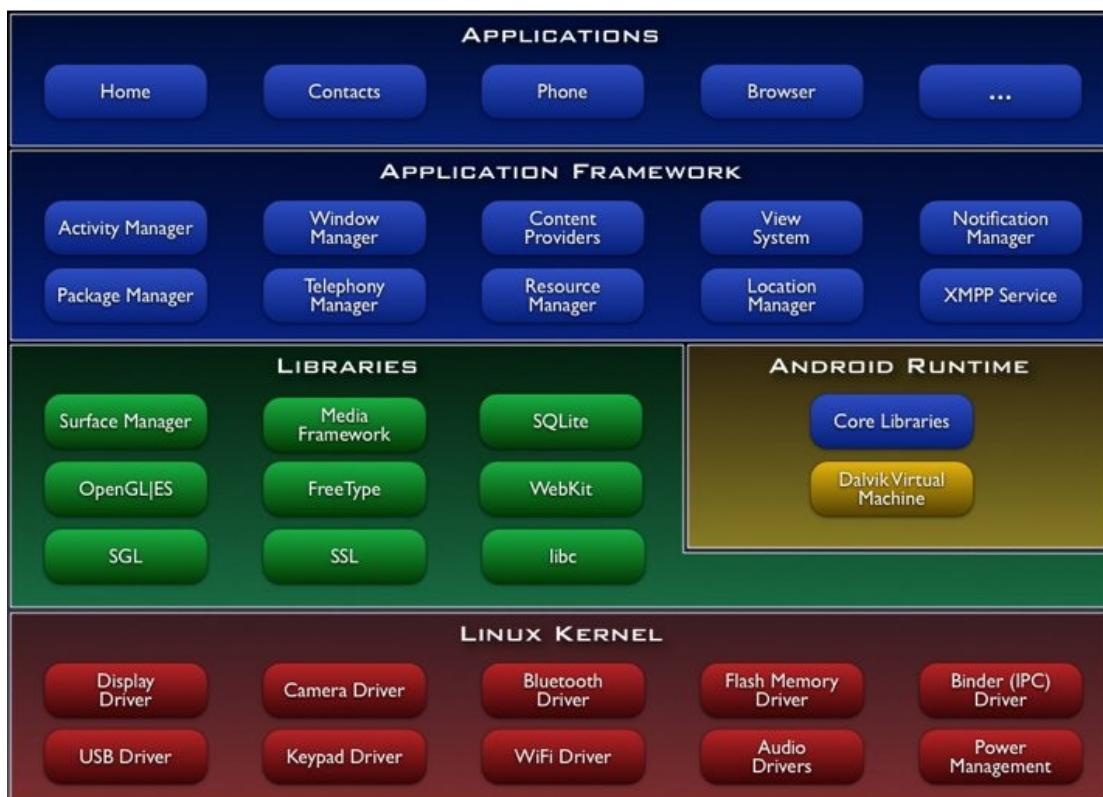
11.2 檔案系統 - Files	89
11.3 資料庫- Sqlite	90
11.4 資料提供者 – ContentProvider	90
第 12 章網路函式庫	91
12.1 網路相關函式庫	91
12.2Socket 程式設計	91
12.3Apache 的 HttpClient	91
12.4Android 的網路函式庫	91
12.5XMPP 網路服務協定	91
12.6WebKit 瀏覽器的控制	97
第 13 章二維繪圖功能	101
13.1Android 的繪圖相關物件	101
13.2 繪製基本圖形	101
13.3 整合練習 - 繪圖版範例	101
第 14 章 2D 動畫功能	106
14.2 變形與轉換	107
	110
第 15 章 3D 動畫 OpenGL	111
15.13D 動畫的基本概念	111
15.23D 模型的設計	111
15.3OpenGL 的範例	111
第 16 章影像與聲音功能	113
16.1 播放內部影音資源	113
16.2 播放外部影音檔案	113
16.3 播放網路上的影音檔案	113
16.4 摷取畫面	113
16.5 錄音與錄影	114
16.6 照相功能	115
第 17 章電話功能 Telephony	122
17.1 電話相關函式庫	122
17.2 擇打電話	122
17.3 取得電話資訊	123
17.4 簡訊功能	123
17.5E-mail 功能	124
第 18 章衛星定位	125
18.1 衛星 GPS 相關函式庫	125
18.2 衛星定位的程式	125
18.3 控制 Google Map 的顯示	125

<u>18.4 在 GoogleMap 上重疊顯示.....</u>	129
<u>第 19 章藍芽功能.....</u>	132
<u>19.1 開放原始碼的藍芽函式庫 - BlueZ.....</u>	132
<u>19.2 搜尋藍芽裝置.....</u>	132
<u>19.3 檔案與物件交換.....</u>	132
<u>19.4 透過藍芽連上 Internet.....</u>	132
<u>第 20 章整合範例.....</u>	133
<u>20.2 檔案瀏覽器.....</u>	133
<u>第 21 章自製視覺化元件.....</u>	133
<u>第 22 章部署與安裝程式.....</u>	134
<u>22.1 建立安裝程式.....</u>	134
<u>22.2 部署程式到 Google Phone 上.....</u>	134
<u>第 23 章 Google Phone 的未來.....</u>	135
<u>23.1 Google</u>	135
<u>23.2 結論.....</u>	135
<u>附錄 1Android 的開發工具</u>	136
<u>附錄 2Eclipse 的使用方法</u>	137
<u>附錄 3 本書範例程式的使用方法.....</u>	137
<u>附錄 4Adb 除錯工具的使用.....</u>	141
<u>附錄 5 設定 Android 中 Proxy 的方法.....</u>	146
<u>附錄 6Android 的函式庫列表.....</u>	147
<u>附錄 7Android 中的視覺畫元件展示館</u>	152
<u>附錄 9Android 說明文件的閱讀指引</u>	156
<u>附錄 10Android 與 Java 之間的關係.....</u>	164
<u>附錄 11 常見的錯誤狀況與解決方式.....</u>	164
<u>附錄 12Android 的內核程式碼.....</u>	167
<u>參考文獻.....</u>	169

第1章 Android 平台簡介

1.1 Android 平台的結構

Android 平台是 Google Phone 的手機作業平台，平台的底層採用了 Linux 作為作業系統（圖一中的紅色部分），在 Linux 作業系統之下、內建了許多控制裝置的驅動程式，包含藍芽(Bluetooth)、無線網路(WiFi)、隨插即用的 USB 介面等等，當然還有記憶體、銀幕、鍵盤、照像、音效等裝置的驅動程式，在 Linux 作業系統之上，內建了許多由 C/C++ 語言所開發出來的函式庫 (Libraries)（圖一中的綠色部份），包含 libc、OpenGL/ES、WebKit、Sqlite 等，接著、Google 架構出 Java 環境 Android Runtime，這包含了一個虛擬機器 Dalvik Virtual Machine (DVM) 與基本的 Java 函式庫 (Core Libraries)，然後、Google 利用這些 C/C++ 函式庫與 Java 虛擬機器建構出一組應用架構 (Application Framework)，以協助應用程式設計者開發出好的應用程式 (Application)。



目前、Android 當中內建的應用程式已有瀏覽器 (Web Browser)、Google Map、手機通話 (Phone) 等，在未來應該會有更多的應用被開發出來，在筆者寫作的同時，Google 也正在舉辦一個 Android 應用軟體大賽，總共提供了 1000 萬美元的獎金，為的應該就是要快速的充實 Android 手機平台上的軟體。

Android 平台的函式庫主要來自下列三個群體，第一個是 Android 本身開發的，第二個是 Java 原有的函式庫，第三個是 開放原始碼社群的函式庫 (Android 採用最多的是從 Apache Software Foundation 來的)

1.2 Google Phone 的外觀與功能

要成為一個 Google Phone 的手機程式設計師之前，應該先熟悉 Google Phone 的手機使用環境，在 Android 平台所提供的預設環境當中，有三種形式的 Google Phone，包含 QVGA-L、QVGA-P、HVGA 等，其中 HVGA 又分成垂直版 HVGA-L 與水平版 HVGA-P 兩種，下表顯示了這些手機的型號、大小與樣式。

skinID	Description	Skin
QVGA-L	320x240, landscape (default)	

QVGA-P

240x320, portrait



HVGA-L

480x320, landscape



HVGA-P

320x480, portrait



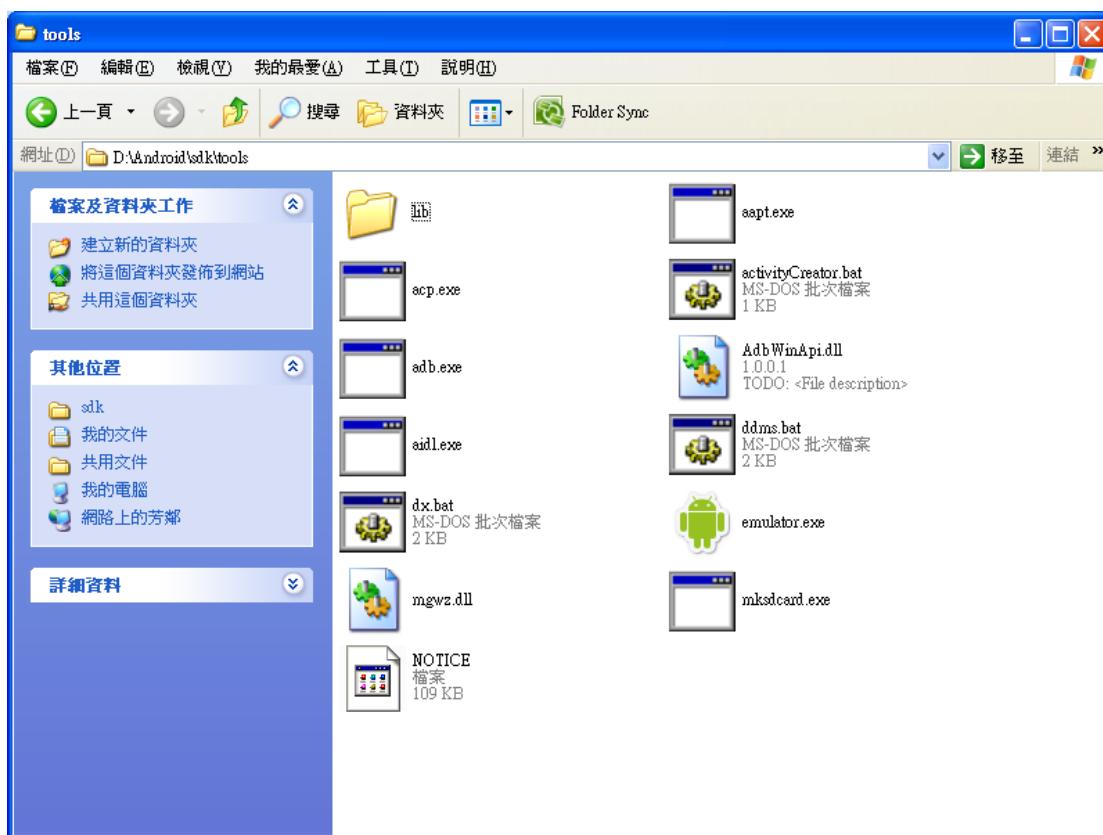


這就是 Google Phone 的樣子了，其中的按鈕大多都很直覺，會使用手機的人應該都很容易上手，然而中間的幾個按鈕式功能鈕，並非一般手機所具有的，下表說明了每個按鈕的功能：

按鈕圖示	按鈕名稱	功能說明
------	------	------

	功能表鈕	按此鈕會顯示功能表 (若該畫面具有功能表的話)
	退回鈕	按此鈕會退回到上一層
	主畫面鈕	按此鈕會回到系統最上層的主畫面
	取消鈕	按此鈕會取消某個動作

1.3 啟動 Google Phone 的模擬器



當我要啟動模擬器時，最簡單的方式就是在 **emulator.exe** 上雙擊滑鼠右鍵，此時、預設的模擬器畫面 QVGA-L 將會被啟動，結果如下圖所示：



在啓動過程完成之後，將會呈現出主視窗畫面如下：

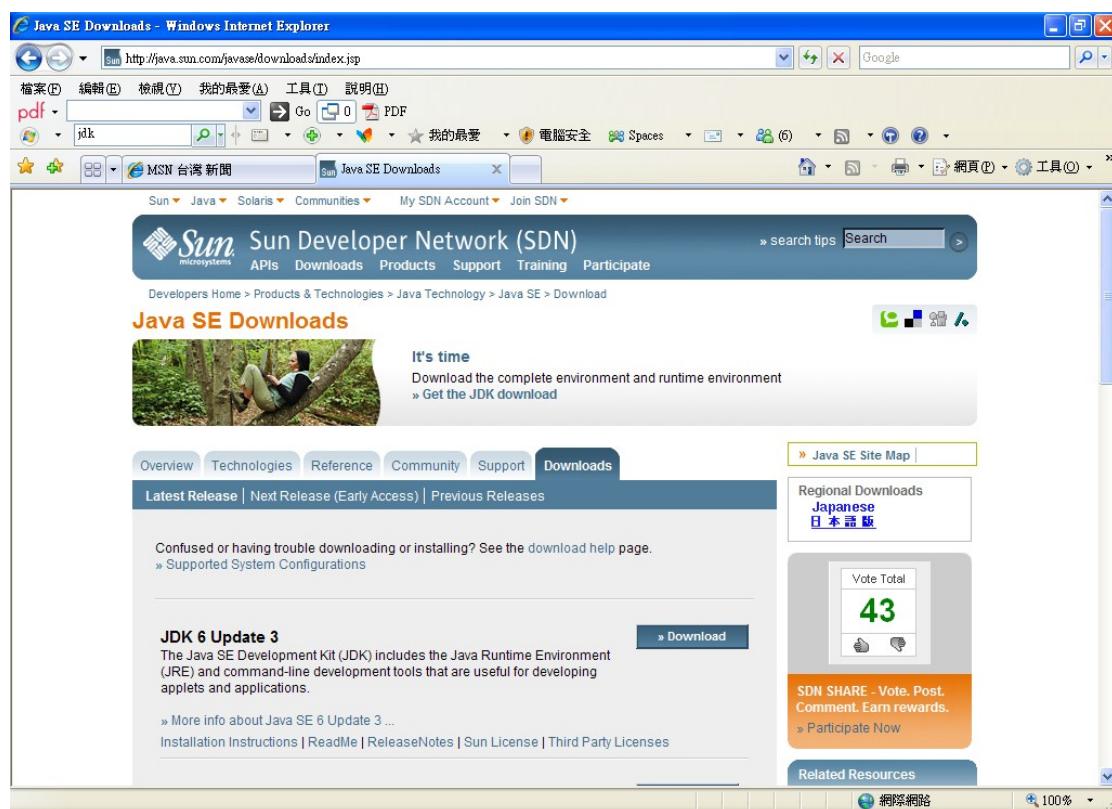


第2章 程式開發環境的安裝

要開發 Google Phone – Android 平台的程式，您必須先安裝 Android 的開發環境，這個環境所採用的平台是建構在 Java 的虛擬機器 Java Virtual Machine (JVM)的一個手機特製版本 Delvik Virtual Machine (DVM) 上的，開發用的程式語言為 Java，並使用 Java 上最主要的整合開發工具 Eclipse 所為開發環境，設計者只要在 Eclipse 當中外掛 Android Development Tools plugin 即可完成開發環境的安裝。

<http://code.google.com/android/intro/installing.html>

2.1 安裝 Java 的 JDK



Download Center - Download - Windows Internet Explorer

https://dlc6d.sun.com/ECom/EComActionServlet?sessionId=7A46013103A6C7D8239106FA6E306AB5

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

pdf - jdk Go 0 PDF

MSN 台灣 新聞 Download Center - Down...

Sun Downloads Search

Download

Java(TM) SE Development Kit 6 Update 3

NOTE: These products are offered as either a single large file or broken up into parts (multiple smaller files). This page offers files for different platforms - please be sure to download the proper file(s) for your platform.

We highly recommend using Sun Download Manager (SDM), as it lets you pause, resume, and restart your download while ensuring a successful download experience. Just select the files you want to download, then click the "Download Selected with Sun Download Manager" button to automatically install and start SDM. Alternately, click directly on the links in the file list to download through your browser.

For any download problems or questions, please see the Download Center FAQ.

How long will the download take?

Required: You must accept the license agreement to download the product.

Accept License Agreement | Review License Agreement
 Decline License Agreement

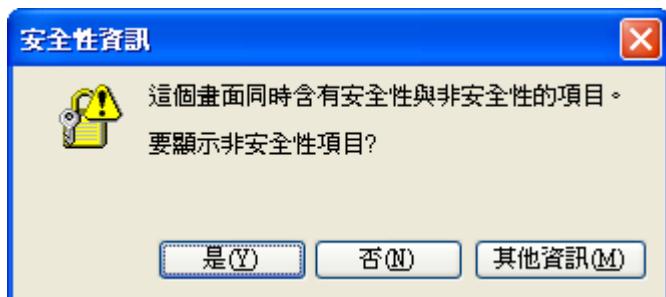
Download selected with Sun Download Manager Easily manage your downloads (pause, resume, restart, verify).
» Learn more

Windows Platform - Java(TM) SE Development Kit 6 Update 3

Download the full version as a single file.

<input type="checkbox"/>  Windows Offline Installation, Multi-language	jdk-6u3-windows-i586-p.exe	65.64 MB
---	----------------------------	----------

正在等候 https://dlc6d.sun.com/ECom/EComActionServlet?sessionId=7A46013103A6C7D8239106FA6E306AB5



Download Center - Download - Windows Internet Explorer

https://sdlc6d.sun.com/ECom/EComActionServlet/DownloadPage~com.sun.sunit.sdlc.content.DownloadPageInfo;sessionId=7A

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

pdf - jdk Go 0 PDF

MSN 台灣 新聞 Download Center - Down...

Sun Downloads Search

Download

Java(TM) SE Development Kit 6 Update 3

NOTE: These products are offered as either a single large file or broken up into parts (multiple smaller files). This page offers files for different platforms - please be sure to download the proper file(s) for your platform.

We highly recommend using Sun Download Manager (SDM), as it lets you pause, resume, and restart your download while ensuring a successful download experience. Just select the files you want to download, then click the "Download Selected with Sun Download Manager" button to automatically install and start SDM. Alternately, click directly on the links in the file list to download through your browser.

For any download problems or questions, please see the Download Center FAQ.

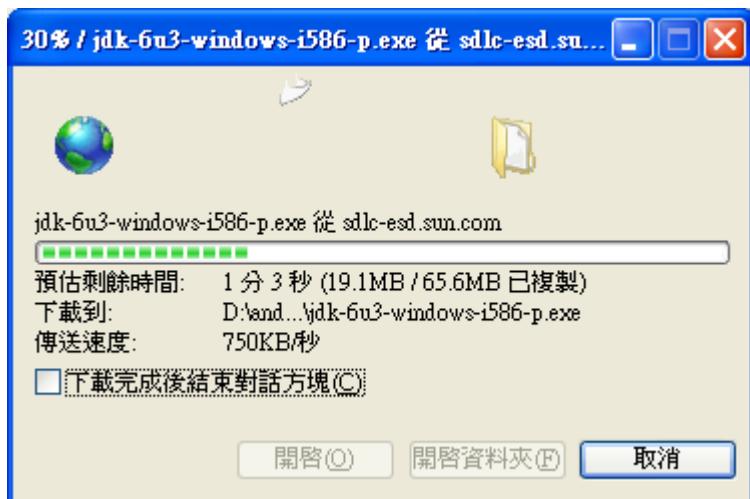
How long will the download take?

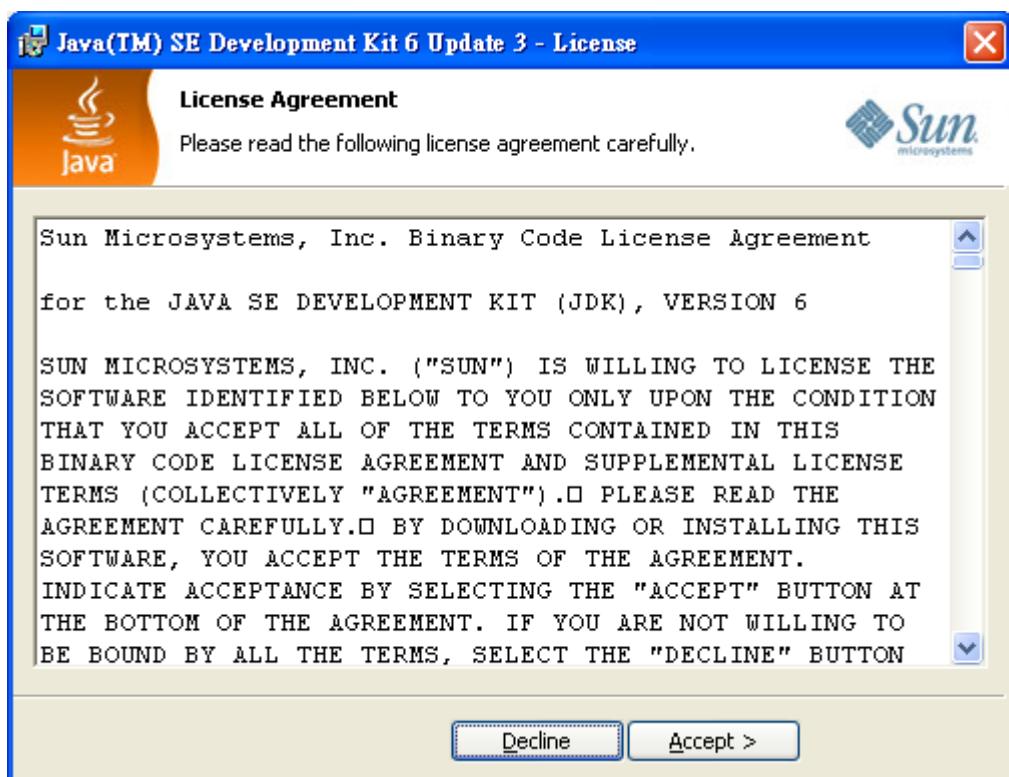
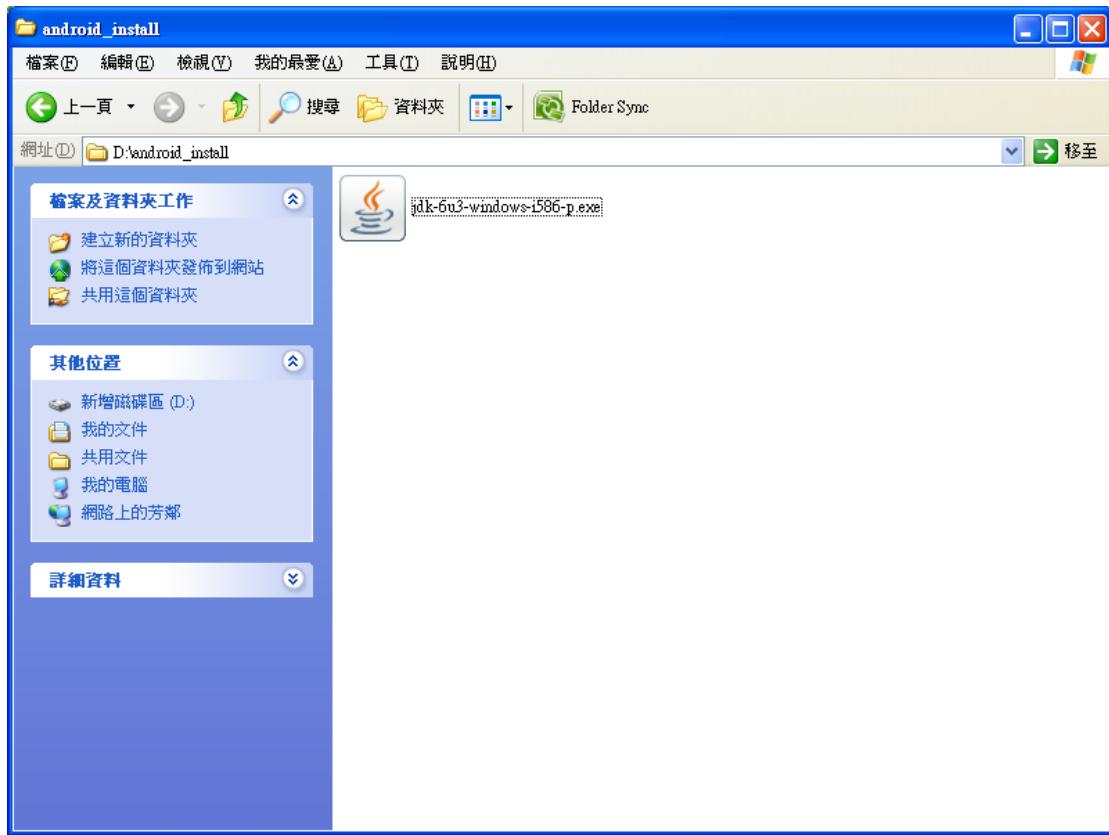
Download selected with Sun Download Manager Easily manage your downloads (pause, resume, restart, verify).
» Learn more

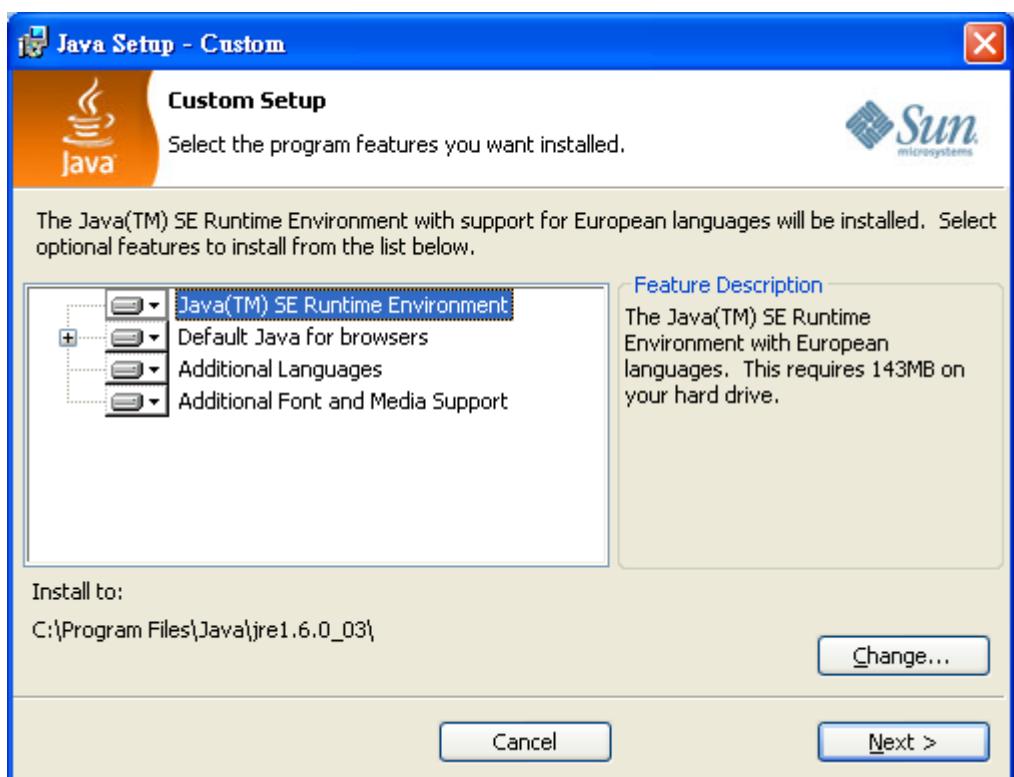
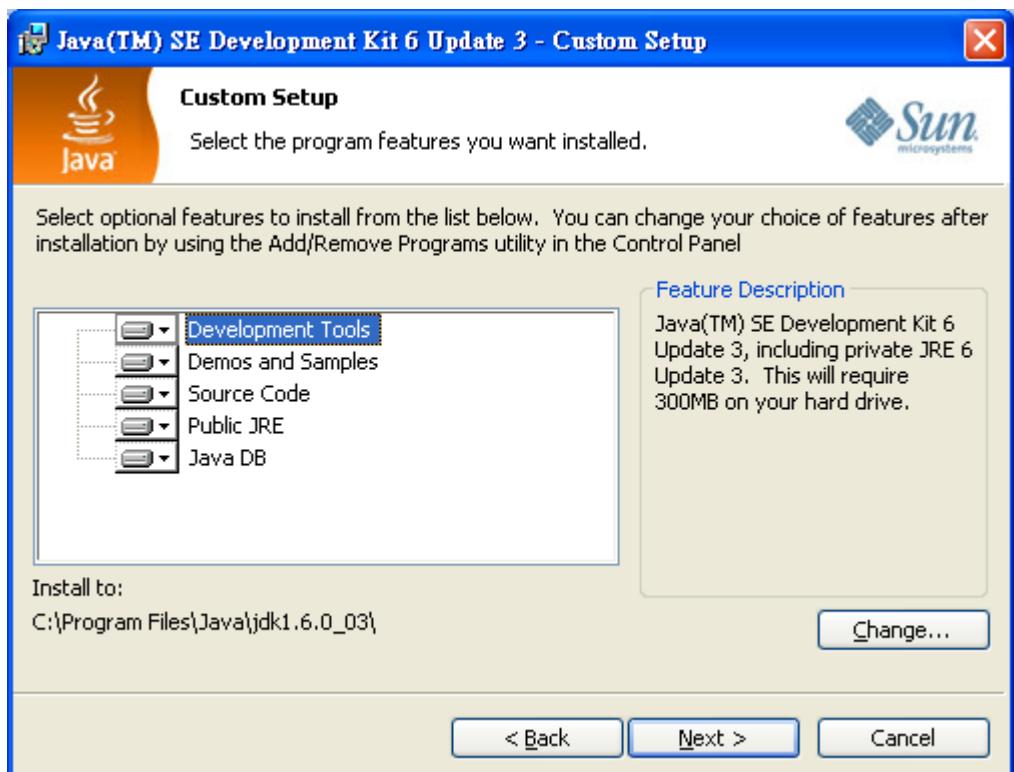
Windows Platform - Java(TM) SE Development Kit 6 Update 3		
<input checked="" type="checkbox"/>	Download the full version as a single file.	
<input checked="" type="checkbox"/>	Windows Offline Installation, Multi-language	jdk-6u3-windows-i586-p.exe 65.64 MB
<input type="checkbox"/>	Windows Online Installation, Multi-language	jdk-6u3-windows-i586-p-iftw.exe 373.39 KB

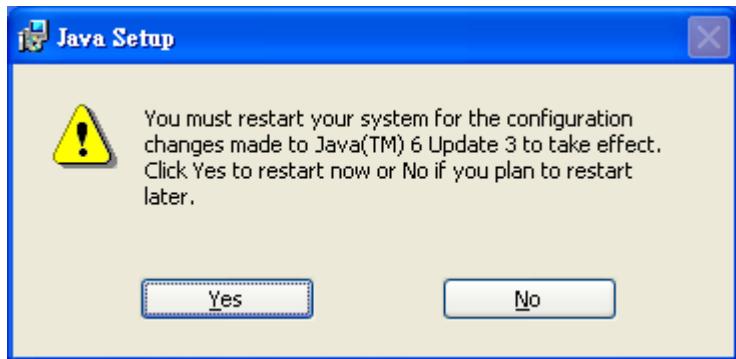
Linux Platform - Java(TM) SE Development Kit 6 Update 3

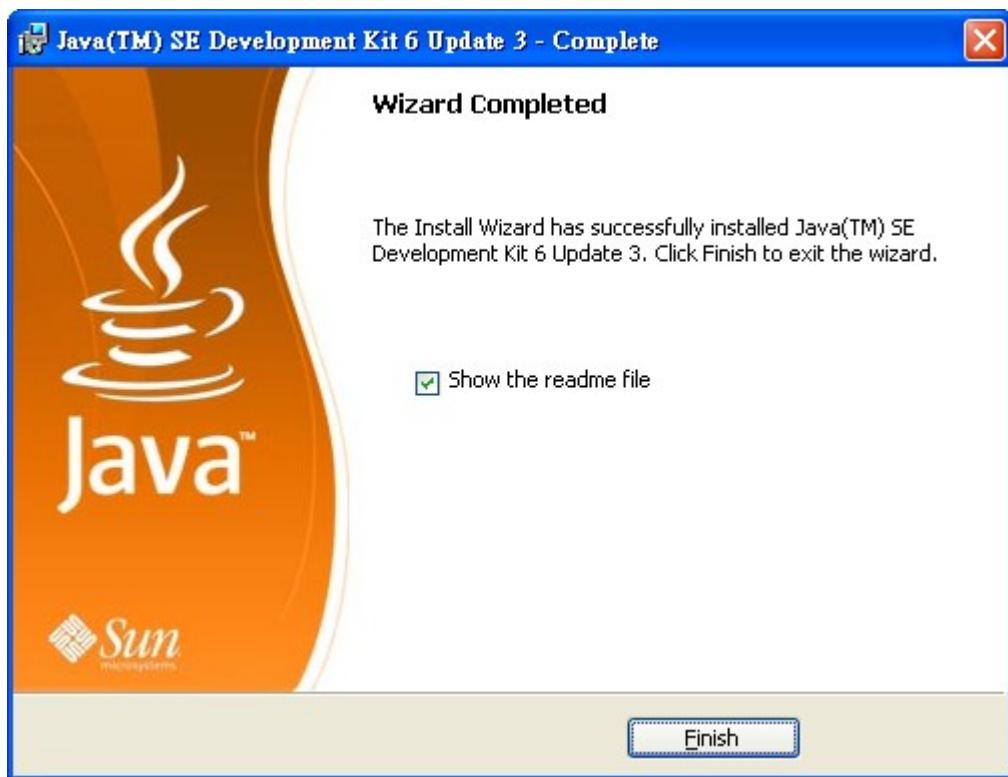












A screenshot of a Microsoft Internet Explorer browser window. The title bar says "README -- Java Platform, Standard Edition Development Kit - Windows Internet Explorer". The address bar shows the URL "C:\Program Files\Java\jdk1.6.0_03\README.html". The page content is the "README" for the Java Platform, Standard Edition Development Kit, version 6. It includes sections for "Contents", "Introduction", and "System Requirements & Installation". A sidebar on the right lists various Java-related links such as "Compatibility", "Bug Reports and Feedback", "Contents of the JDK", etc.

2.2 安裝 Android 的 SDK

Download the Android SDK - Android - Windows Internet Explorer

http://code.google.com/android/download.html

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
pdf - android Go PDF

MSN 台灣 新聞 Download the Android SD... Site Directory

e.g. "intent" or "activity"

ANDROID

Android - An Open Handset Alliance Project Home Docs FAQ Blog Group Terms

[Documentation](#)
[What is Android?](#)
[Getting Started](#)
[Developing](#)
[Applications](#)
[Developer Toolbox](#)
[Reference](#)
[Information](#)
[Sample Code](#)
[FAQs](#)
[Goodies](#)

Download the Android SDK

Before downloading, please read the [Terms](#) that govern your use of the Android SDK.

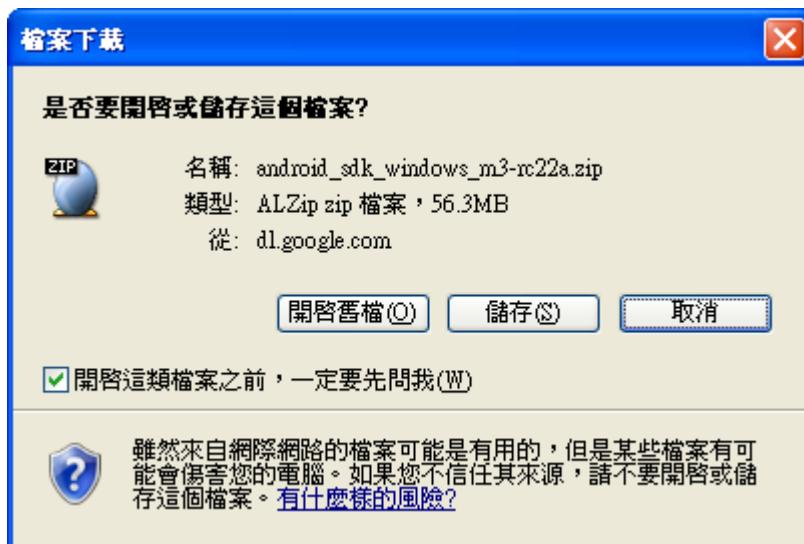
[Download previous versions](#)

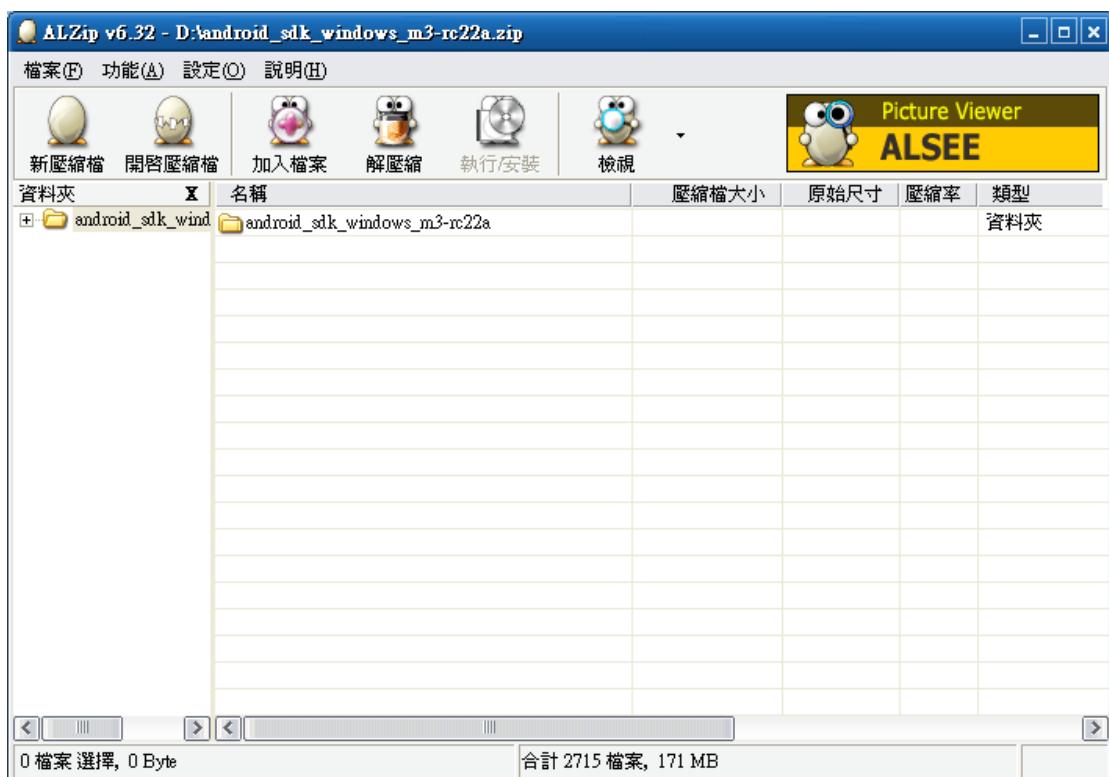
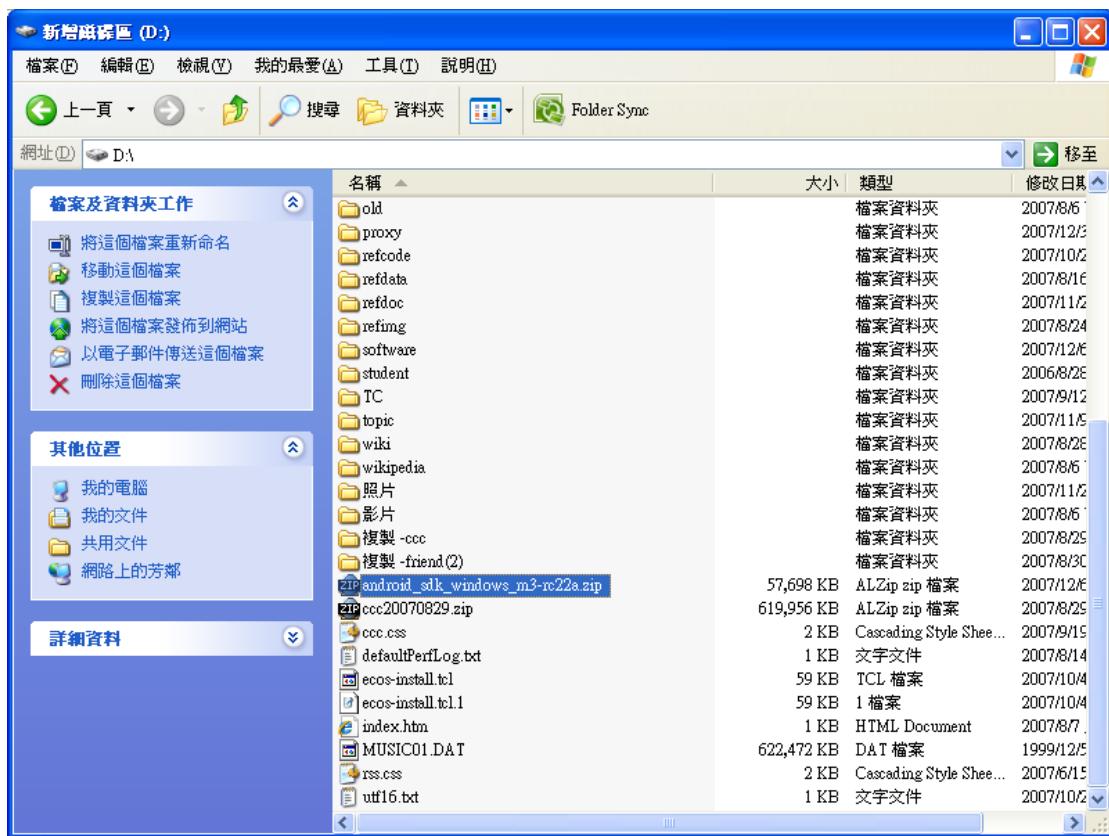
Version m3-rc22a

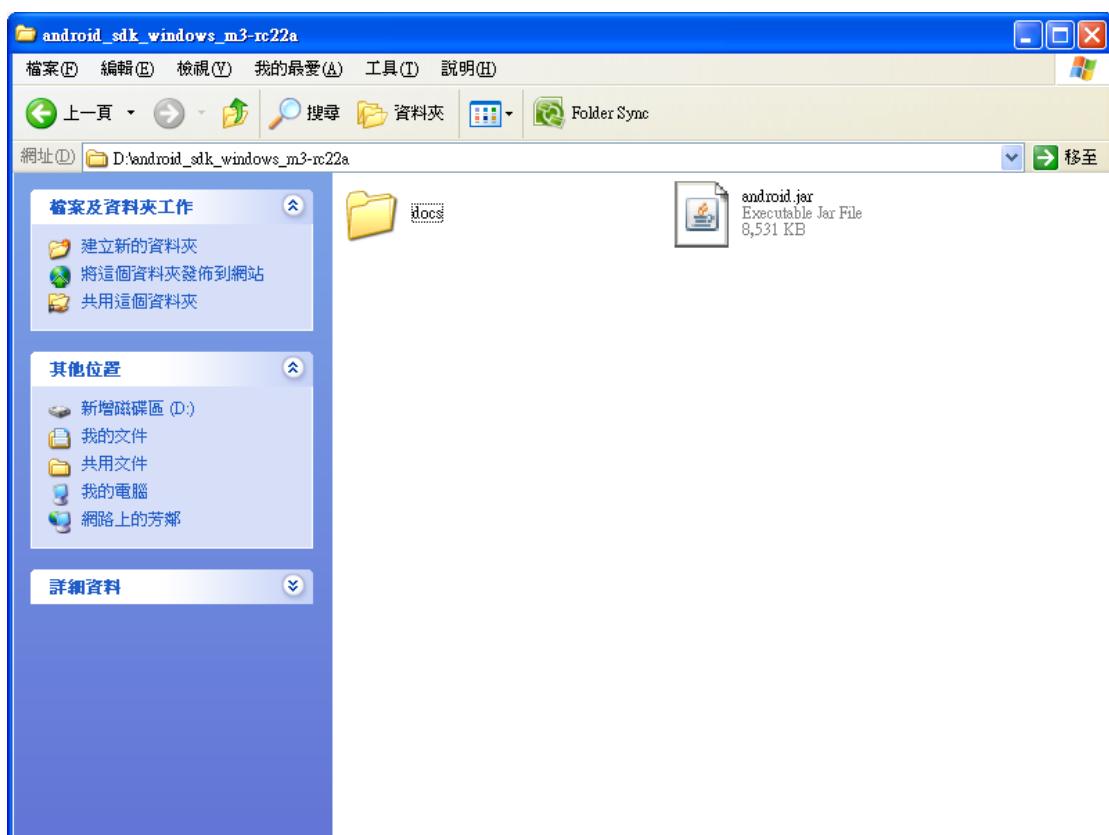
November 16, 2007 - [Release Notes](#)

Platform	Package	Size	MD5 Checksum
Windows	android_sdk_windows_m3-rc22a.zip	59 MB	aa3dee05a9872752a3bc4efd0f93e98b
Mac OS X (intel)	android_sdk_darwin_m3-rc22a.zip	55 MB	0547f45614ad94c3af22c3c0aa6f709f
Linux (i386)	android_sdk_linux_m3-rc22a.zip	55 MB	84b3455de5cdfd841a172c13d24c382e

點選 windows 中的 android_sdk_windows_xxx.zip









2.3 安裝 Eclipse 整合開發環境

Eclipse.org home - Windows Internet Explorer

http://www.eclipse.org/

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

pdf Go 0 PDF

Eclipse Spaces

Eclipse.org home

eclipse

CONTACT | LEGAL

HOME | COMMUNITY | MEMBERSHIP | COMMITTERS | DOWNLOADS | RESOURCES | PROJECTS | ABOUT US

SEARCH | A A

Eclipse - an open development platform

Eclipse is an open source community whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. A large and vibrant ecosystem of major technology vendors, innovative start-ups, universities, research institutions and individuals extend, complement and support the Eclipse platform. New to Eclipse?

Screenshots

Download Eclipse

Eclipse is used for ...

Enterprise Development, Embedded + Device Development, Rich Client Platform, Application Frameworks, Language IDE

Announcements

Now you can make a donation to Eclipse +2 days ago

Remember to submit a proposal for EclipseCon 2008; Deadline is November 19. +3 weeks ago

Results of the Eclipse Community Survey +4 weeks ago

Spotlights

Webinar: Introduction to Eclipse eRCP

Learn more about eRCP in a webinar with Gorkem Ercan on Dec. 13.

Attend an Eclipse DemoCamp

Donate to Eclipse

Useful Links

- Bugs
- Documentation
- Newsgroups
- IRC
- Eclipse Project Wiki

Eclipse Downloads - Windows Internet Explorer

http://www.eclipse.org/downloads/

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

pdf Go 0 PDF

Eclipse Spaces

Eclipse Downloads

eclipse

CONTACT | LEGAL

HOME | COMMUNITY | MEMBERSHIP | COMMITTERS | DOWNLOADS | RESOURCES | PROJECTS | ABOUT US

SEARCH | A A

Eclipse Downloads

To download Eclipse, select a package below or choose one of the third party Eclipse distros. You will need a Java runtime environment (JRE) to use Eclipse (Java 5 JRE recommended). All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Problems extracting the ZIP file? Please read these Known Issues.

Eclipse Europa Fall Maintenance Packages - Windows (compare packages)

Icon	Name	Size	Platform
	Eclipse IDE for Java Developers	78 MB	Windows Linux Mac OS X
	Eclipse IDE for Java EE Developers	126 MB	Windows Linux Mac OS X
	Eclipse IDE for C/C++ Developers	63 MB	Windows Linux Mac OS X
	Eclipse for RCP/Plug-in Developers	153 MB	Windows Linux Mac OS X
	Eclipse Classic 3.3.1.1	140 MB	Windows

Donate to Eclipse

Browse downloads

- Bit Torrents
- By Project
- By Topic
- Source code

Popular projects

- Visual Editor (VE) ↗
- Web Tools ↗
- PHP Development (PDT) ↗
- Modeling Tools (MDT)
- C/C++ Development (CDT) ↗
- Modeling Framework (EMF) ↗
- Business

Eclipse downloads - mirror selection - Windows Internet Explorer

http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.3.1.1-200710231652/eclipse-SDK-3.3.1.1-v

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

pdf Go PDF

Eclipse 電腦安全 Spaces

Eclipse downloads - mirror selection

eclipse

CONTACT | LEGAL

HOME | COMMUNITY | MEMBERSHIP | COMMITTERS | DOWNLOADS | RESOURCES | PROJECTS | ABOUT US

SEARCH A A

Filter

- HTTP mirrors only (xml)
- FTP mirrors only (xml)
- All mirrors (xml)

Eclipse downloads - mirror selection

Please select a mirror for eclipse-SDK-3.3.1.1-win32.zip

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download from: [Taiwan] National Sun Yat-sen University (http)
...or pick a mirror site below.

Friends of Eclipse Mirror

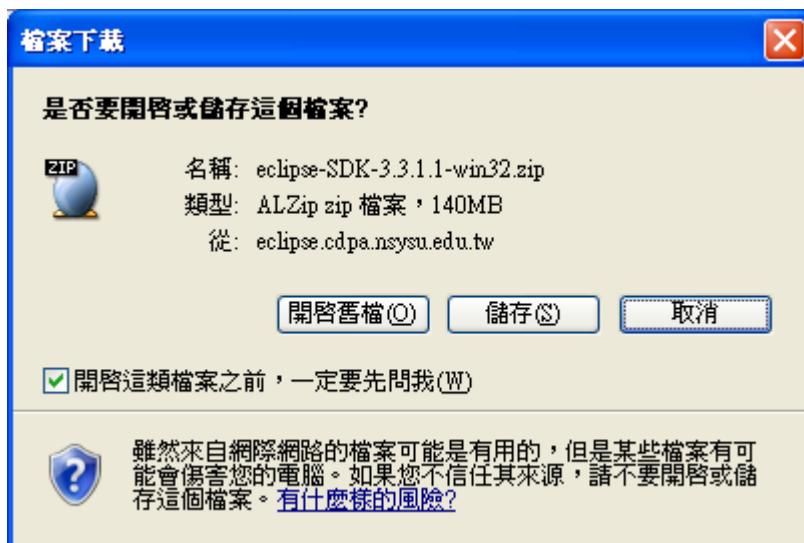
Canada

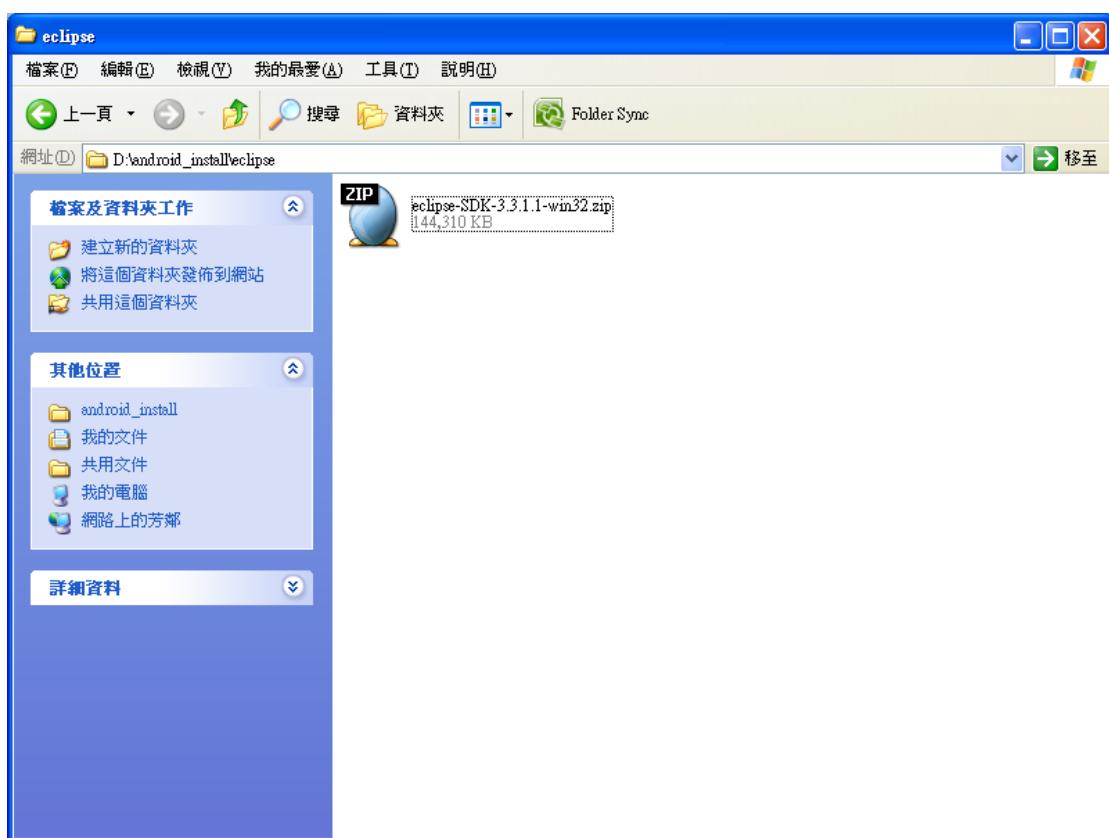
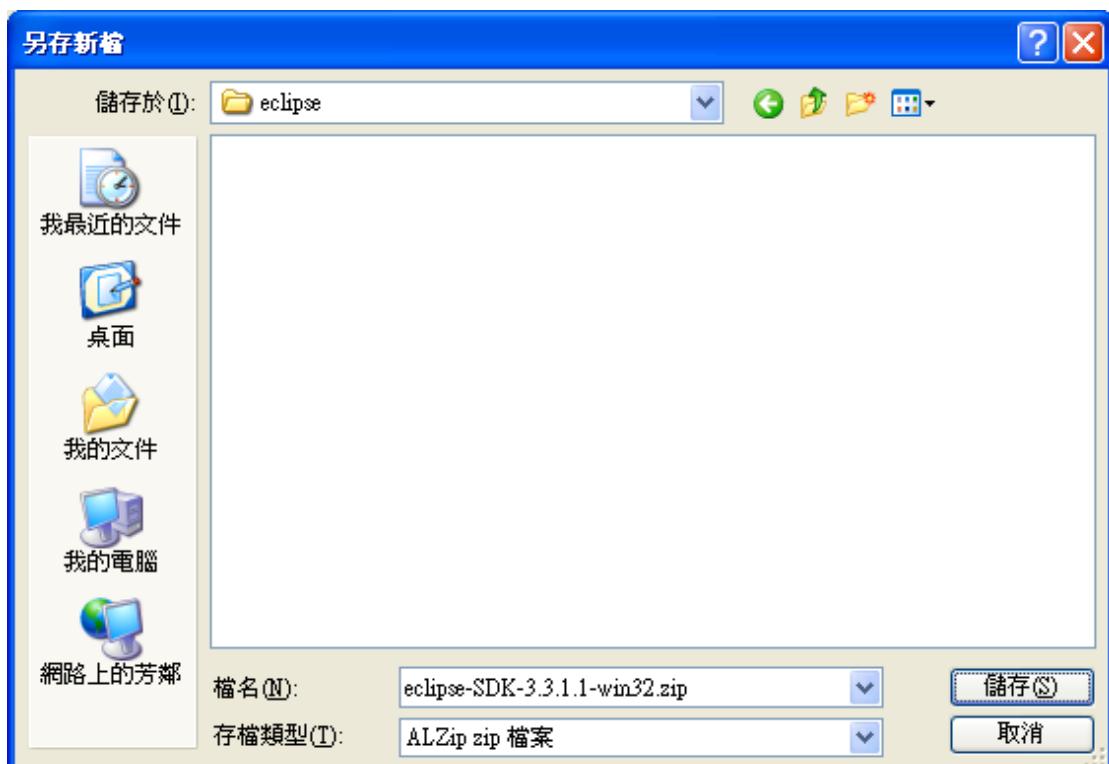
* Friends of Eclipse Mirror * Become a Friend! * Friends login

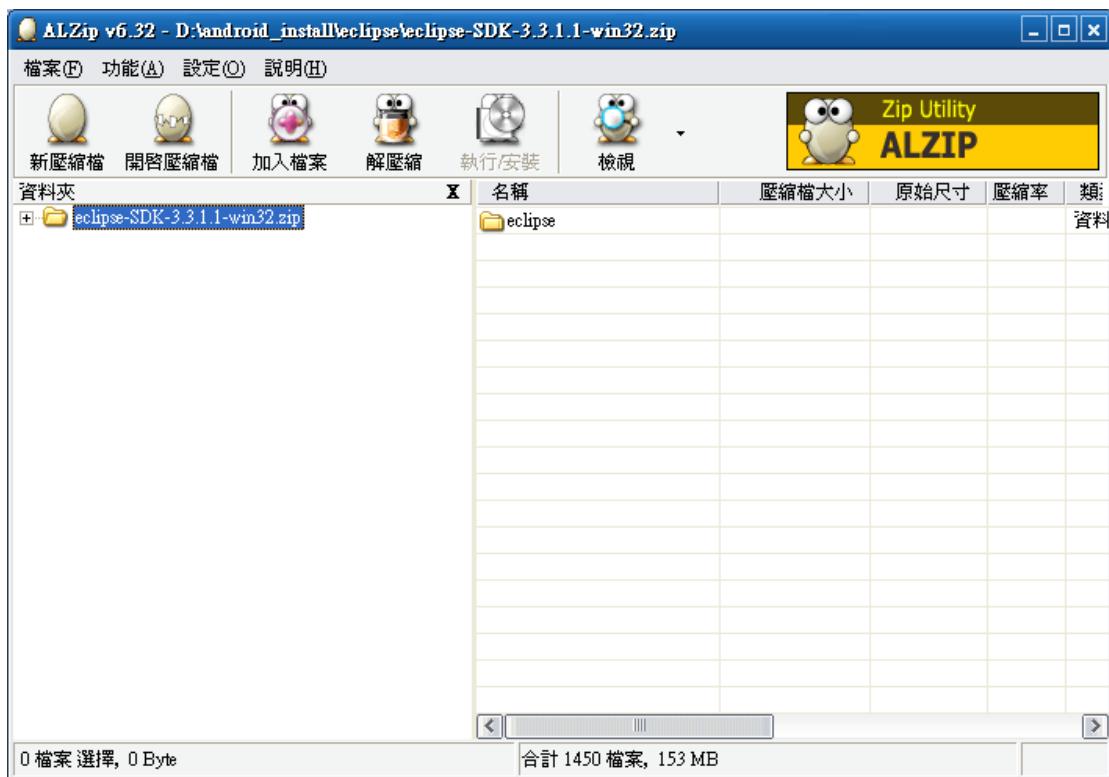
Please choose a mirror close to you

Asia

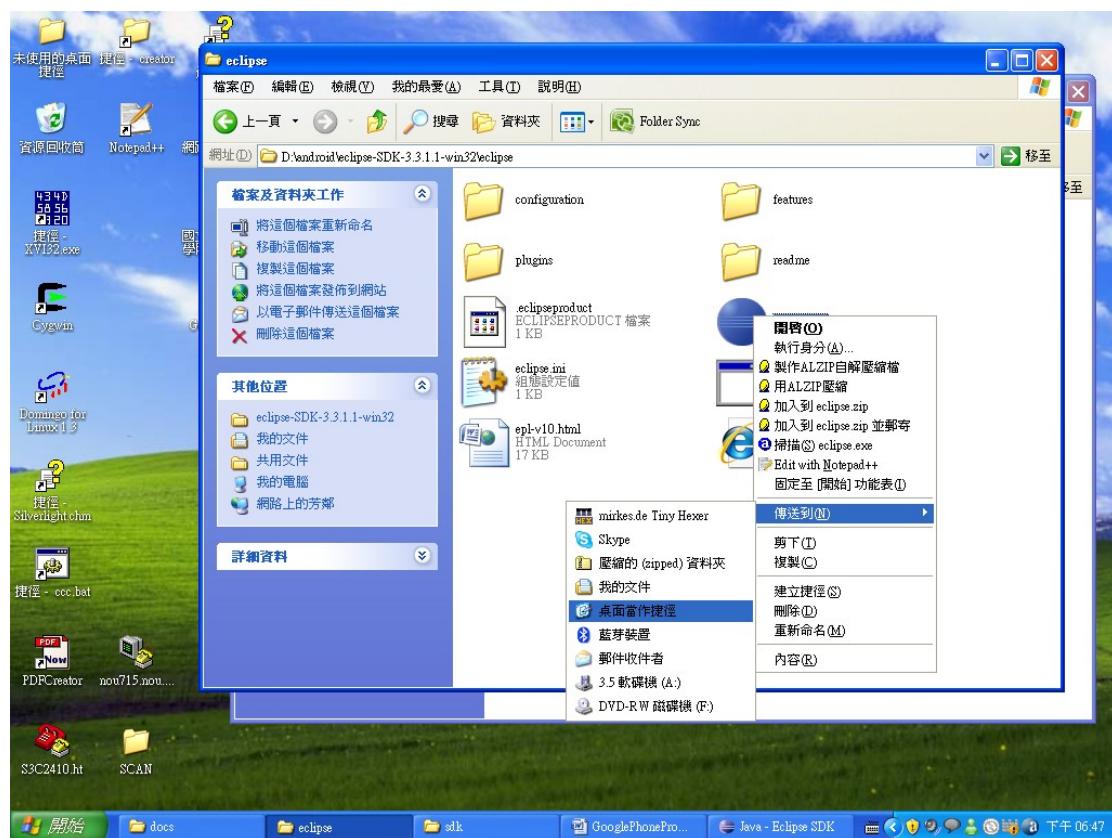
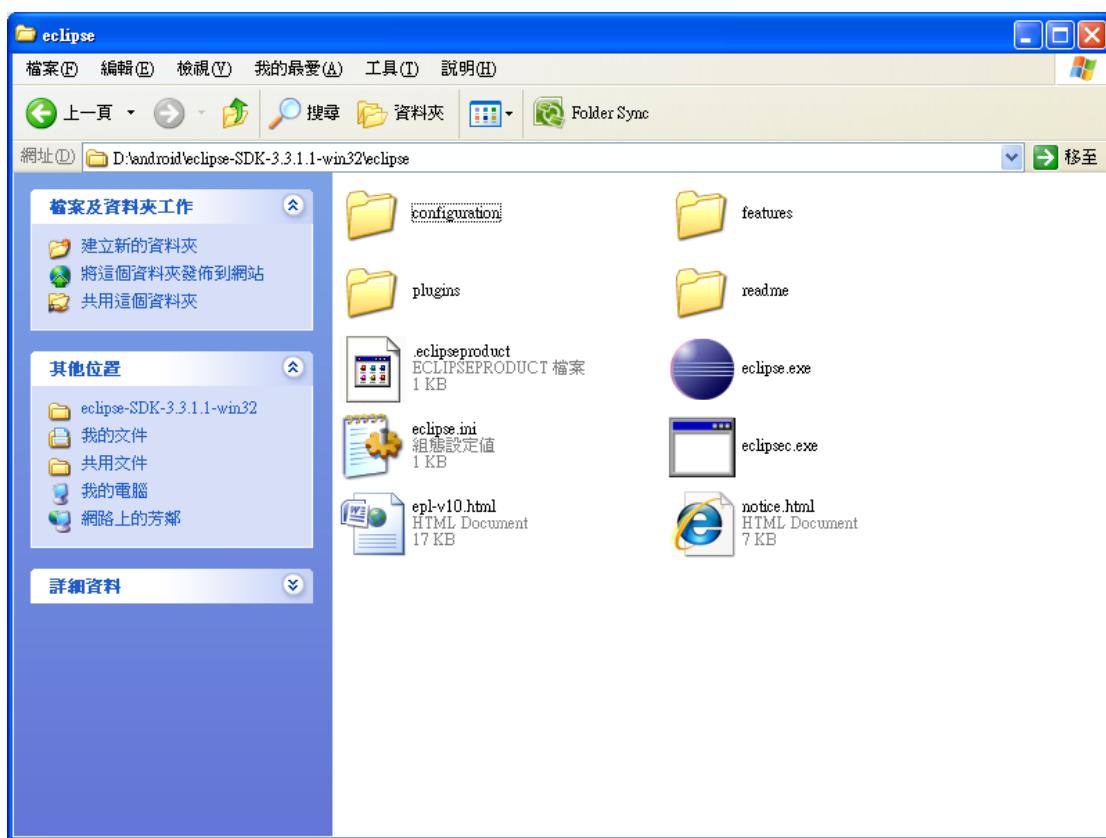
→ [Taiwan] Computer Center, Shu-Te University (http)
→ [Taiwan] National Center for High-Performance Computing (http)
→ Israel NSA Internet & Security I rd. (htm)

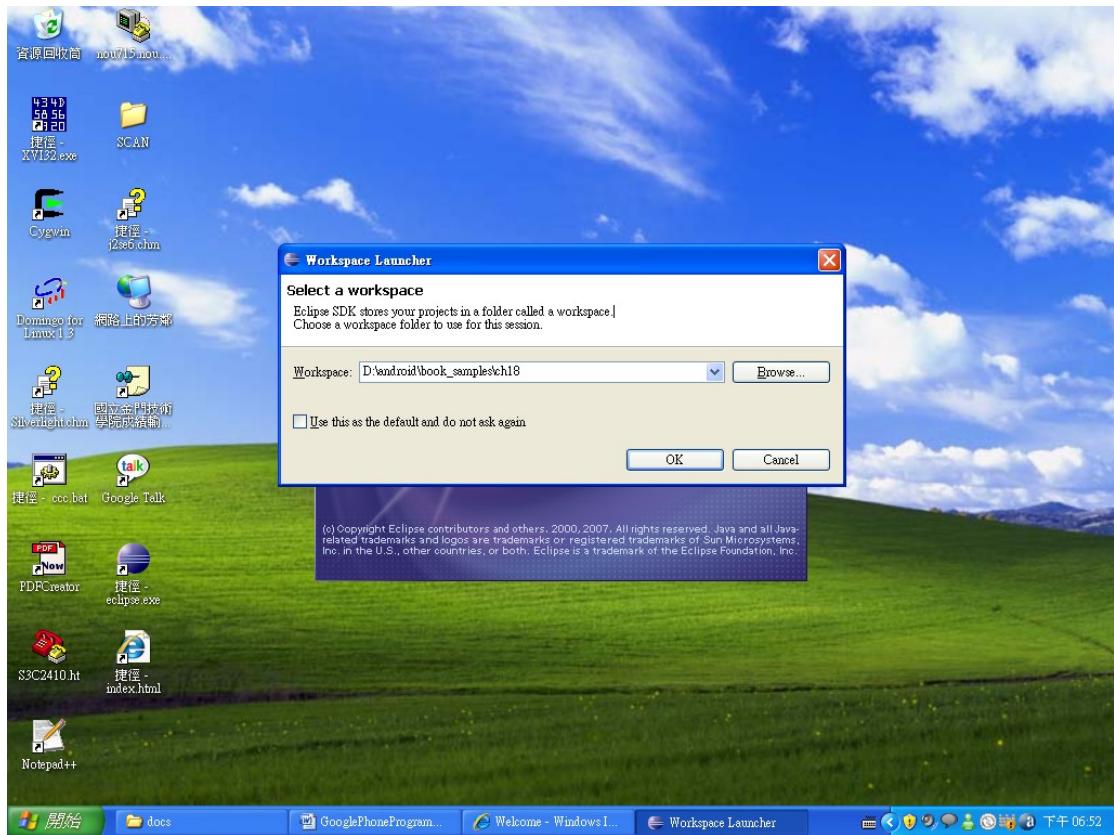






2.4 在 Eclipse 中安裝 Android 外掛元件





Start Eclipse, then select Help > Software Updates > Find and Install....

In the dialog that appears, select Search for new features to install and press Next.

Press New Remote Site.

In the resulting dialog box, enter a name for the remote site (e.g. Android Plugin) and enter this as its URL: <https://dl-ssl.google.com/android/eclipse/>. Press OK.

You should now see the new site added to the search list (and checked). Press Finish.

In the subsequent Search Results dialog box, select the checkbox for Android Plugin > Eclipse Integration > Android Development Tools and press Next.

Read the license agreement and then select Accept terms of the license agreement, if appropriate. Press Next.

Press Finish.

The ADT plugin is not signed; you can accept the installation anyway by pressing Install All.

Restart Eclipse.

After restart, update your Eclipse preferences to point to the SDK

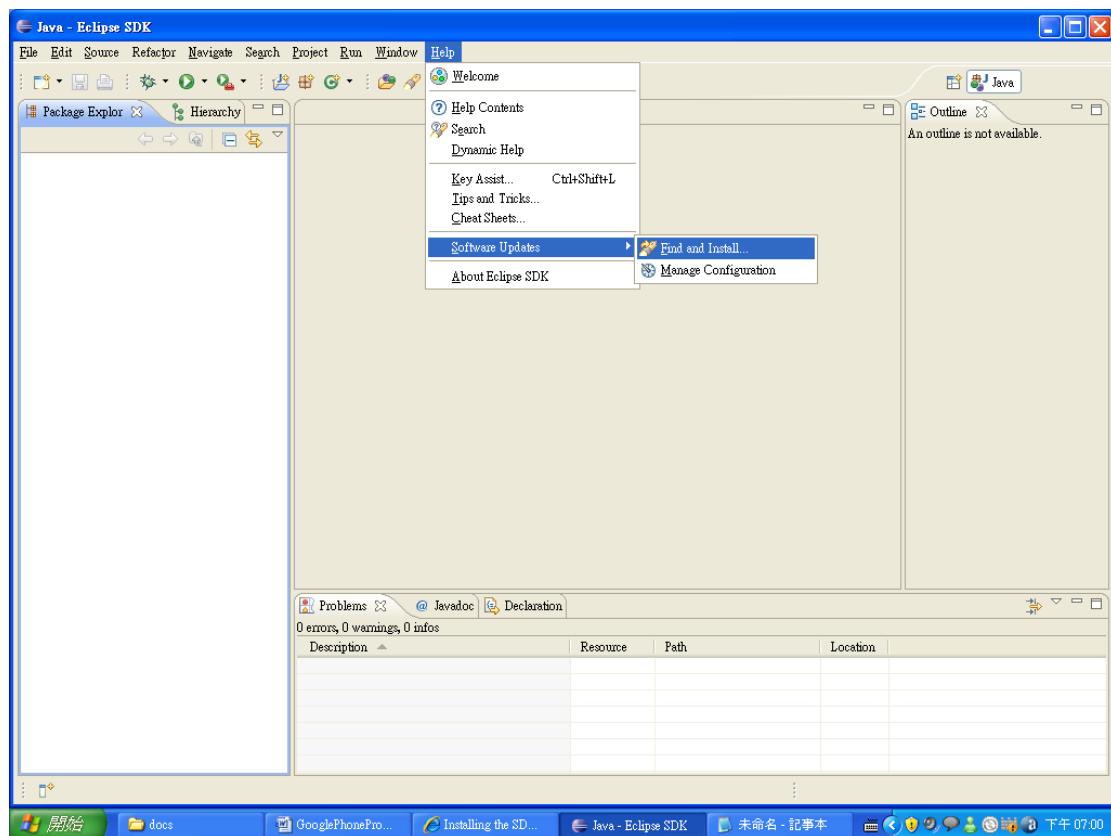
root directory (\$SDK_ROOT):

Select Window > Preferences... to open the Preferences panel. (Mac OS X: Eclipse > Preferences)

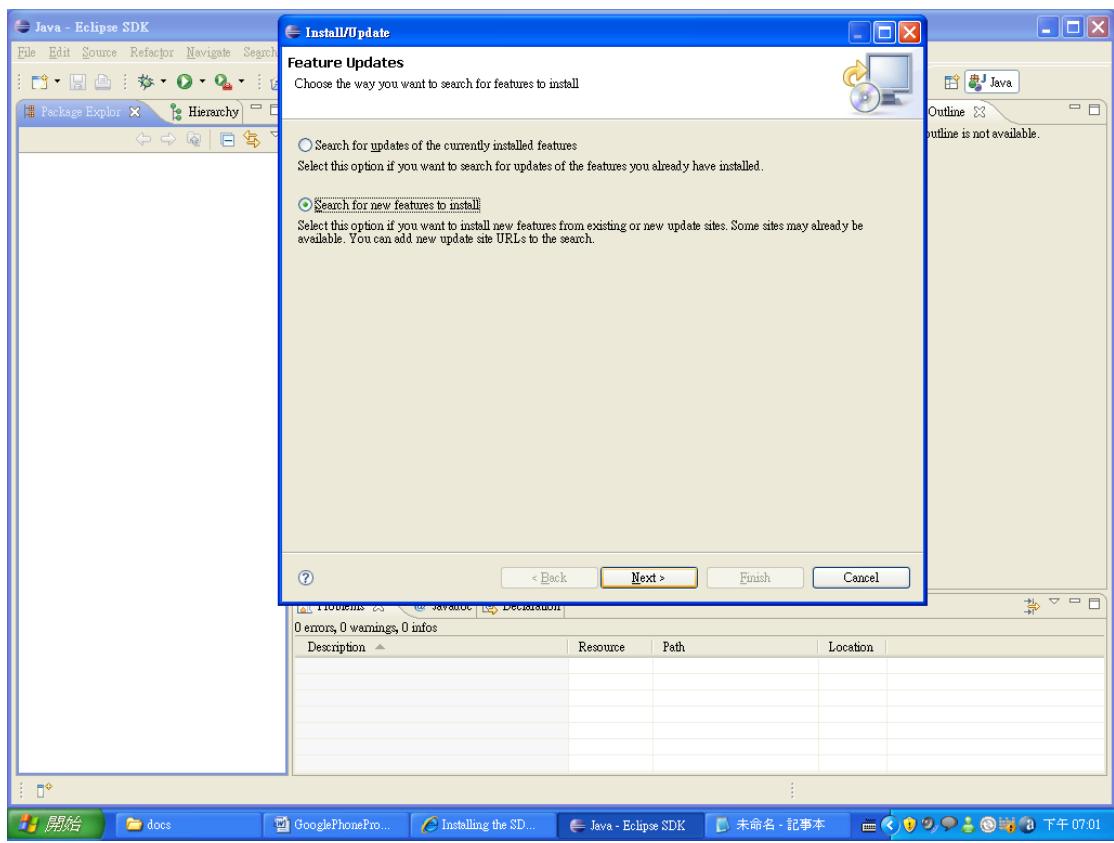
Select Android from the left panel.

For the SDK Location in the main panel, press Browse... and find the SDK root directory.

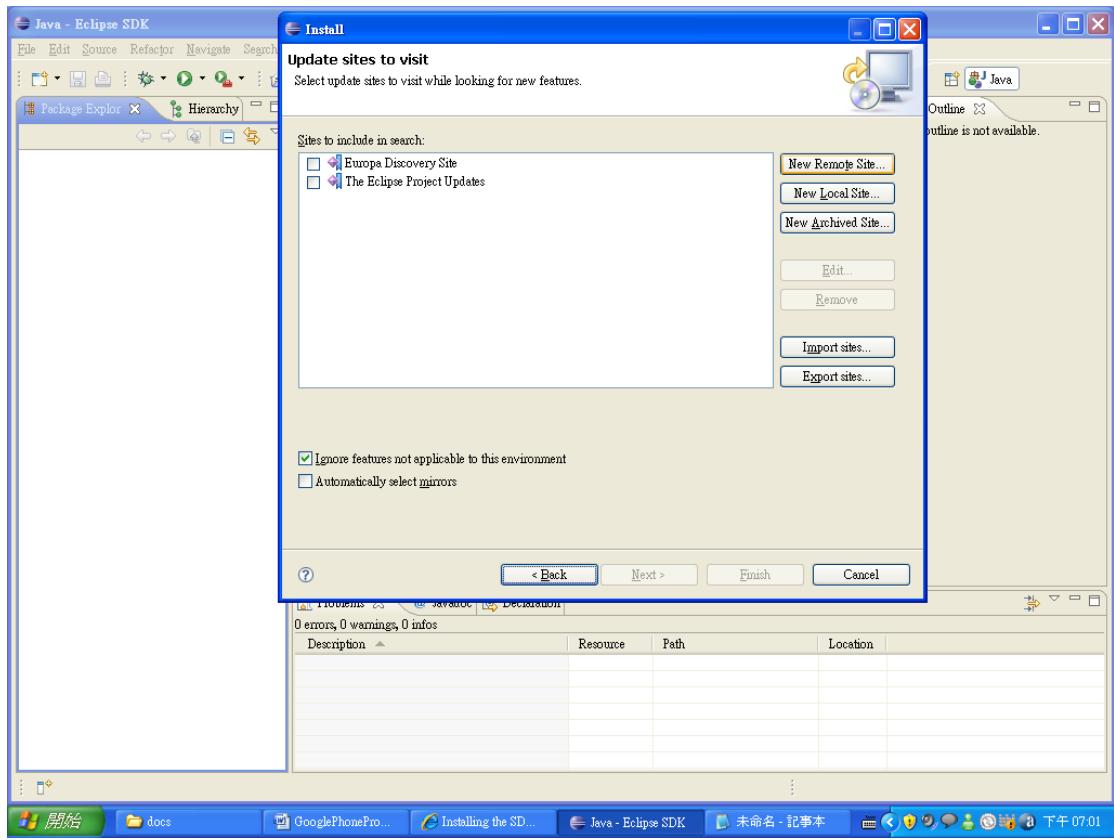
Press Apply, then OK



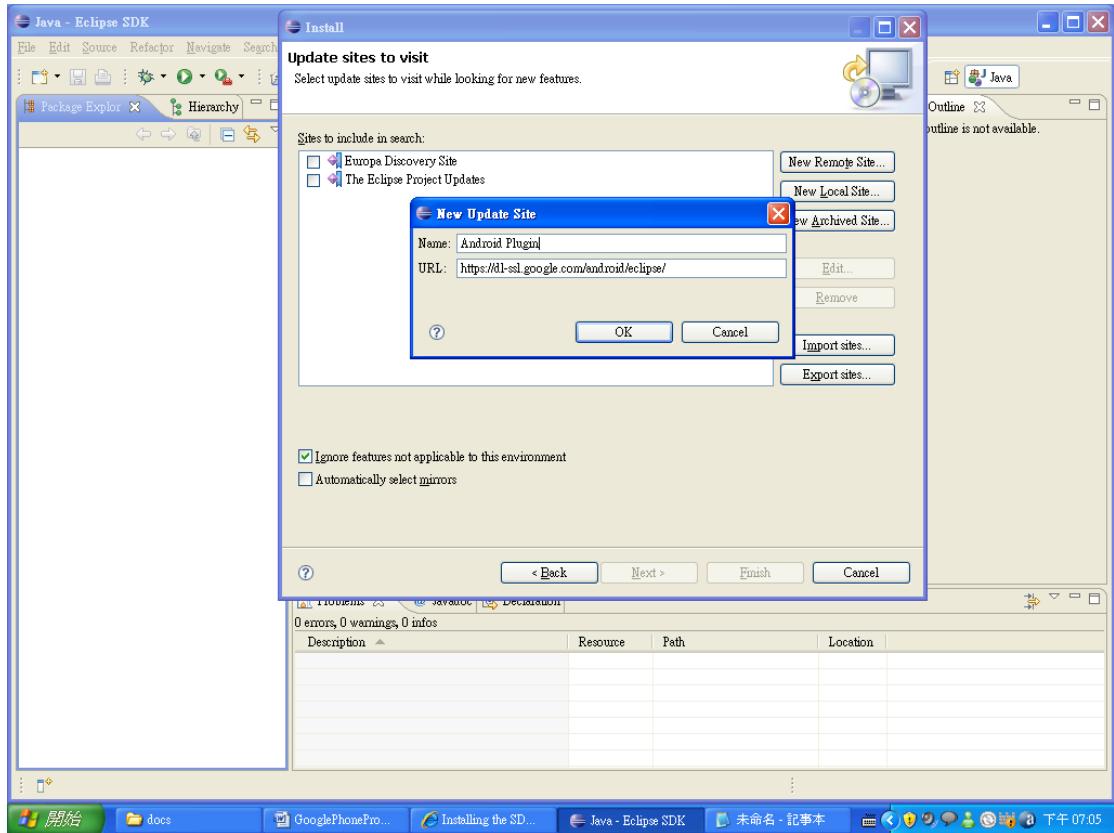
Click Help/Software/Find and Install



Click Search for New Feature to Install

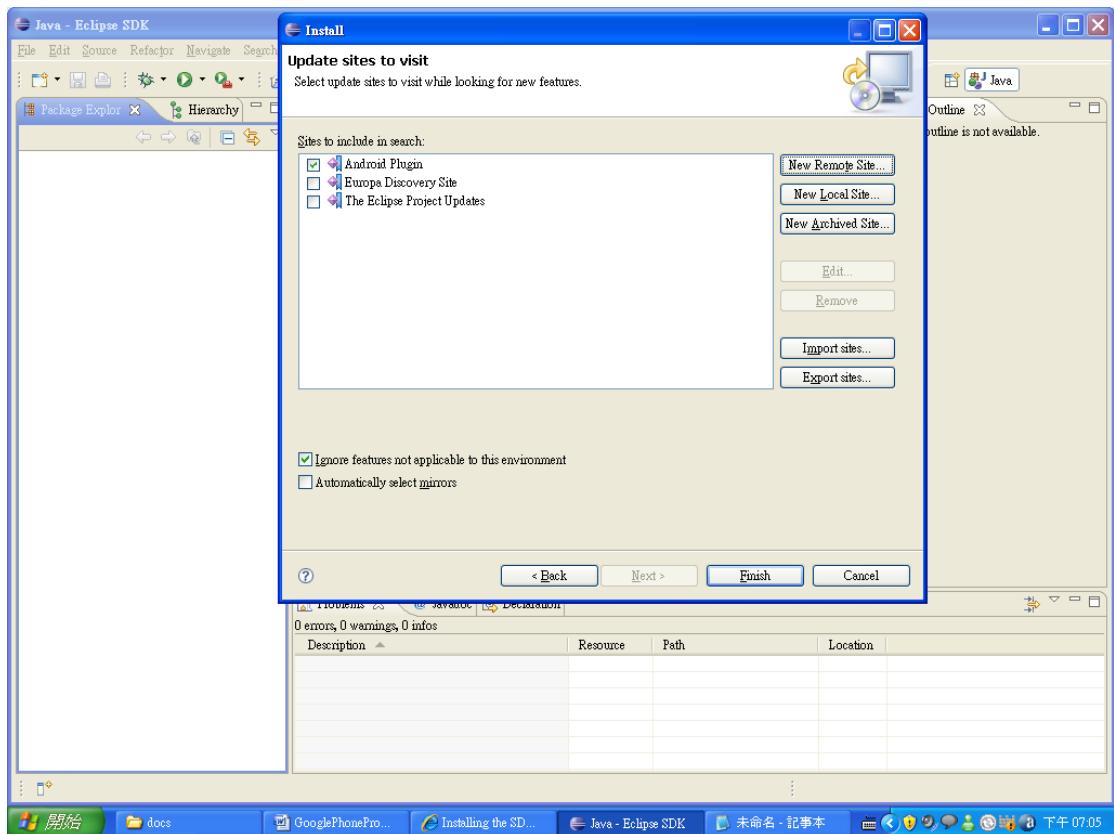


Press New Remote Site

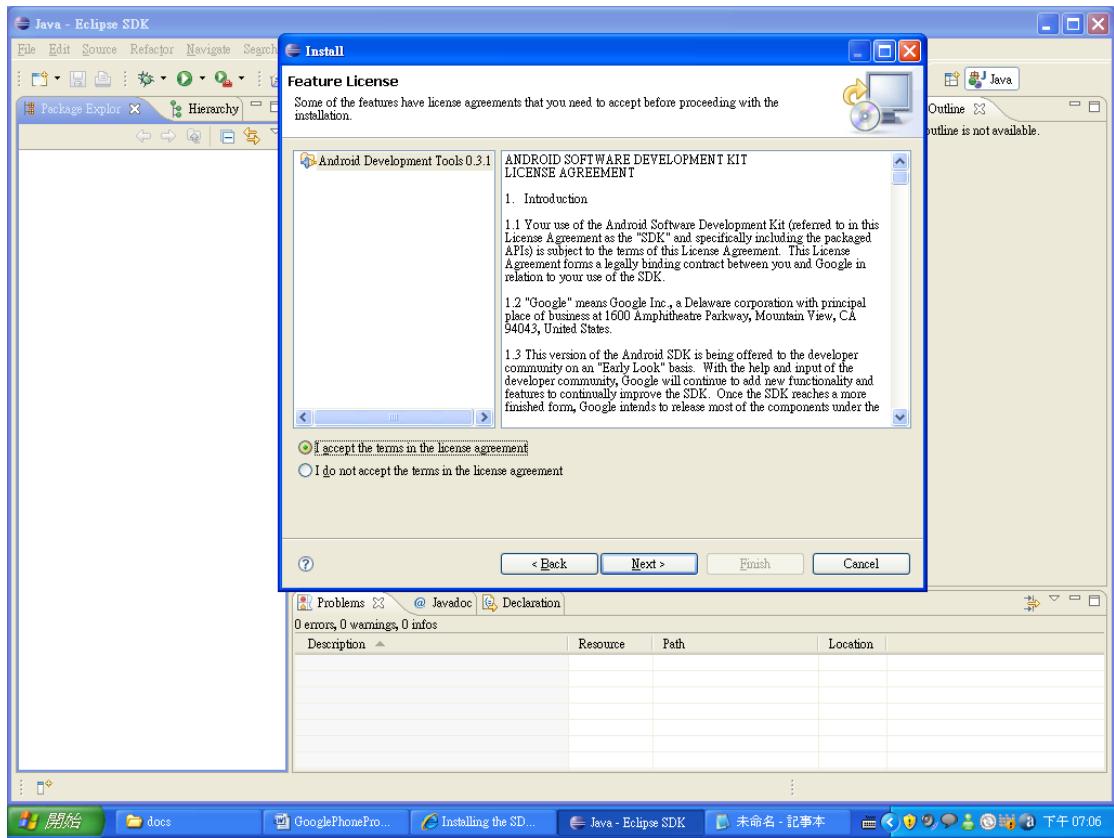
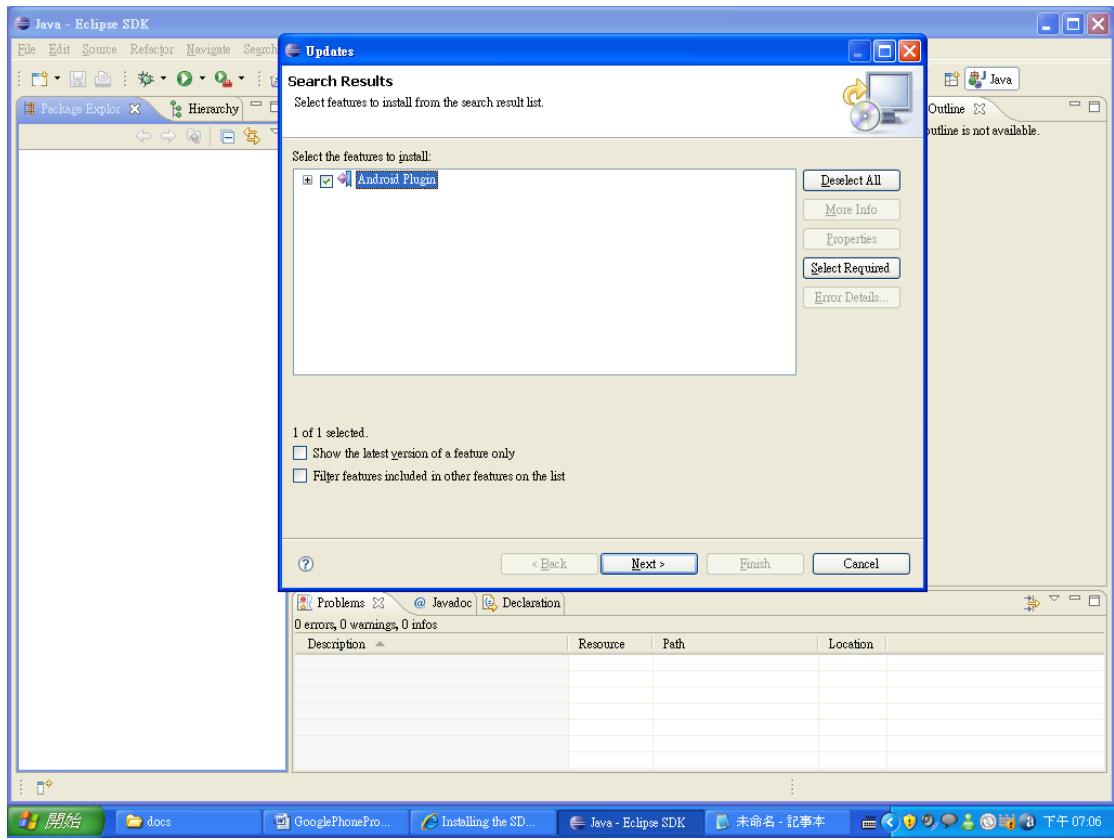


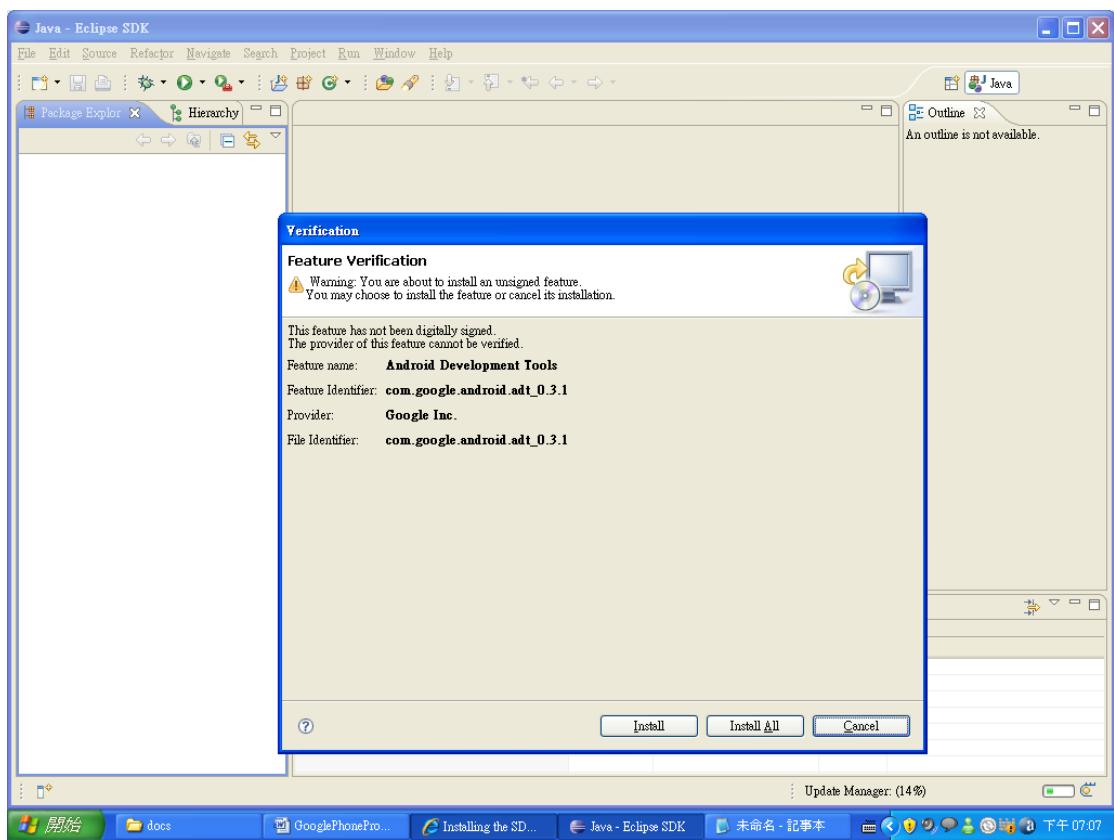
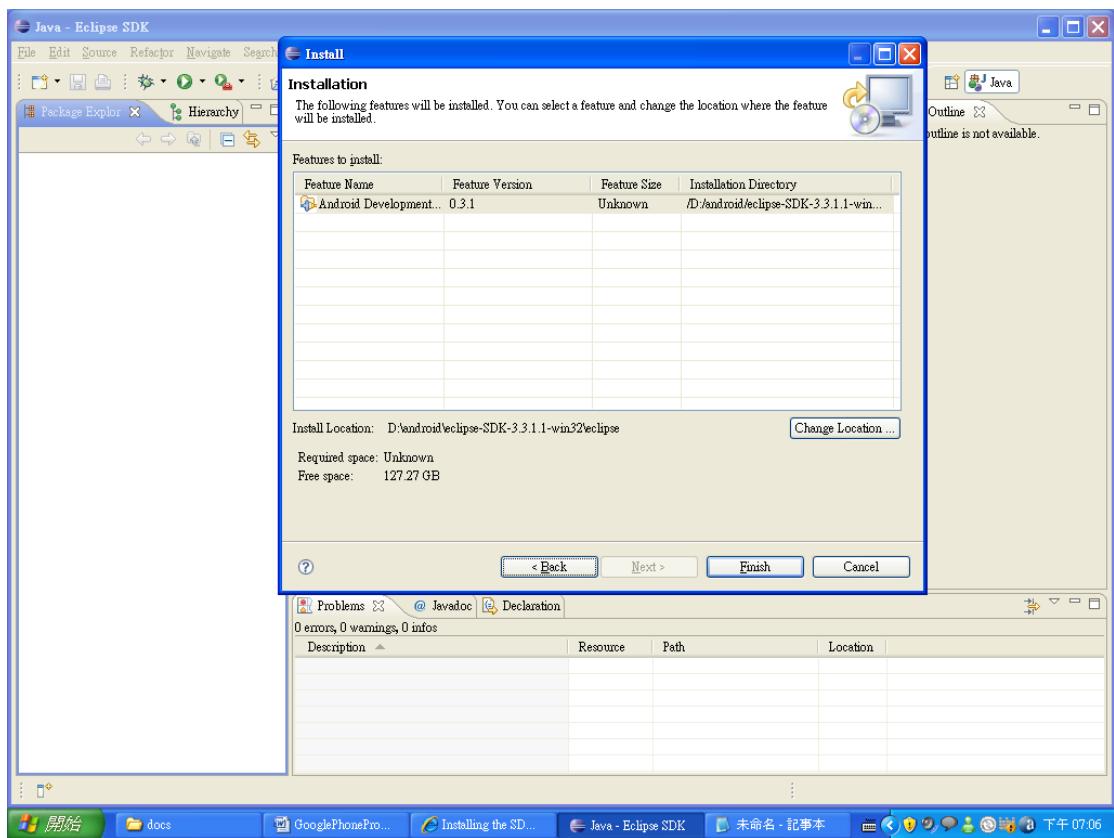
In the resulting dialog box, enter a name for the remote site (e.g.

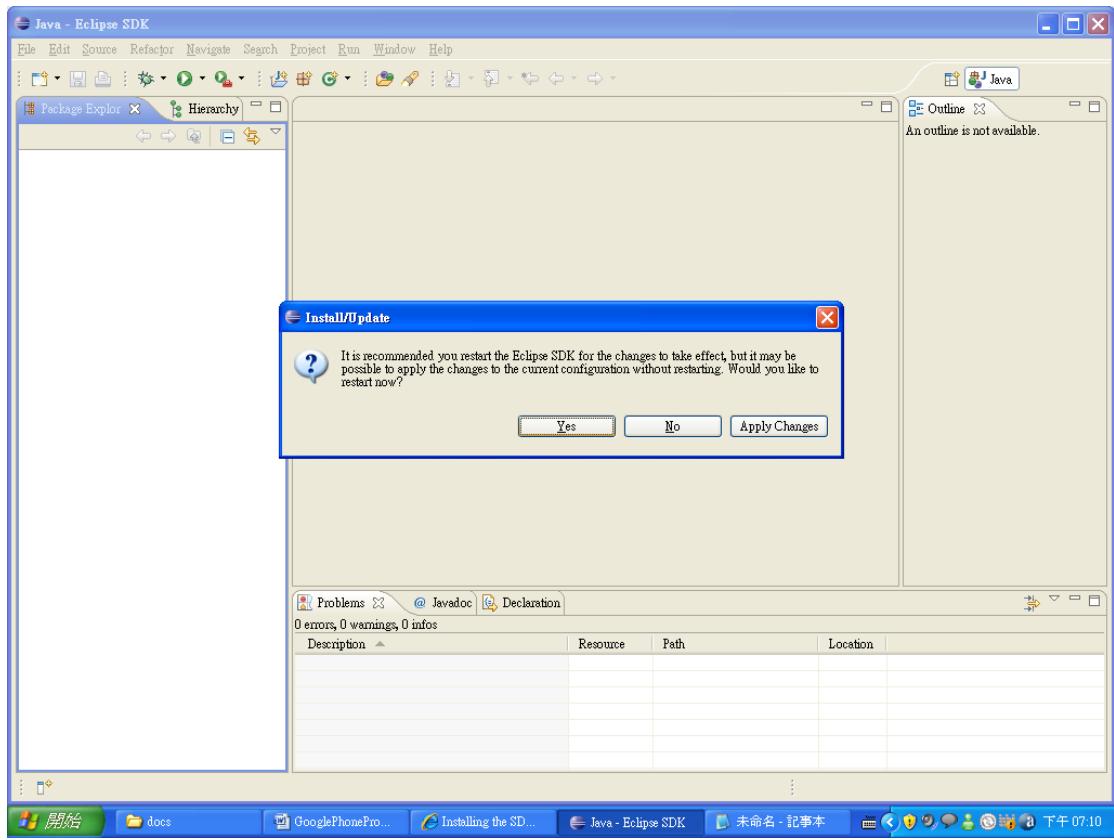
Android Plugin) and enter this as its URL: <https://dl-ssl.google.com/android/eclipse/>. Press OK.



In the subsequent Search Results dialog box, select the checkbox for Android Plugin > Eclipse Integration > Android Development Tools and press Next.







Read the license agreement and then select Accept terms of the license agreement, if appropriate. Press Next.

Press Finish.

The ADT plugin is not signed; you can accept the installation anyway by pressing Install All.

Restart Eclipse.

After restart, update your Eclipse preferences to point to the SDK root directory (\$SDK_ROOT):

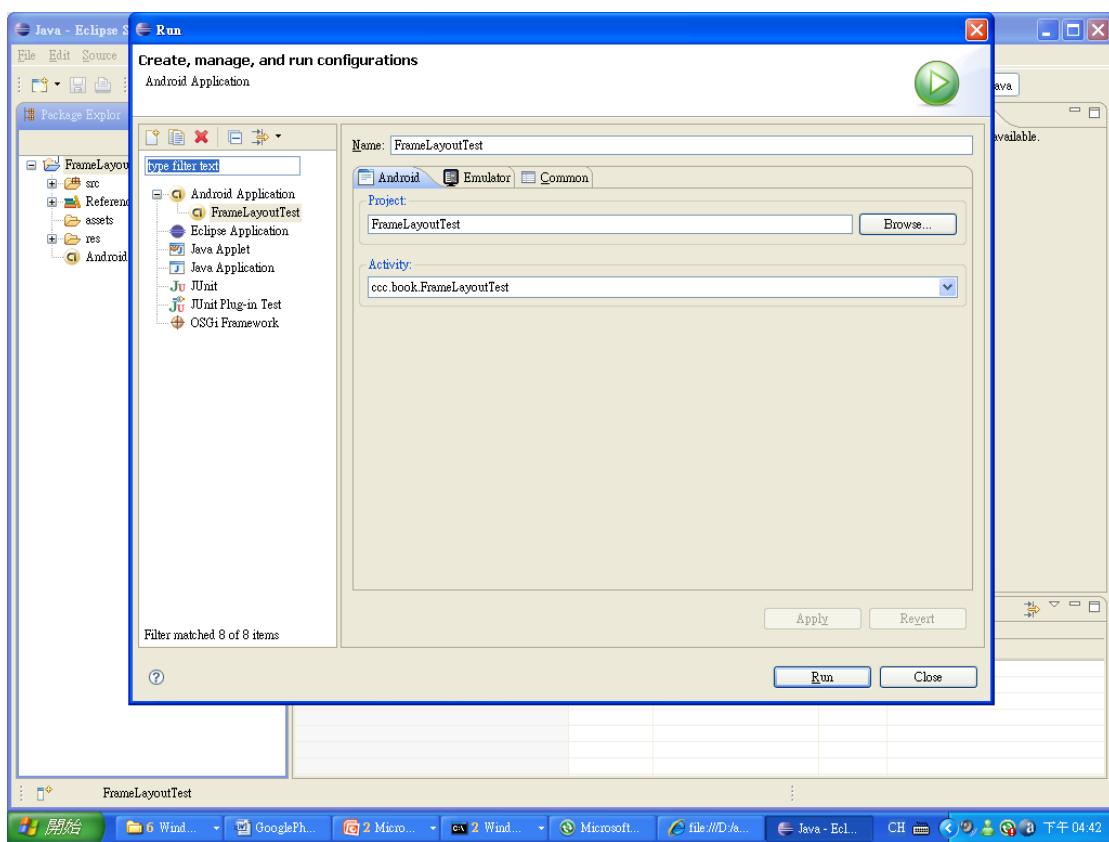
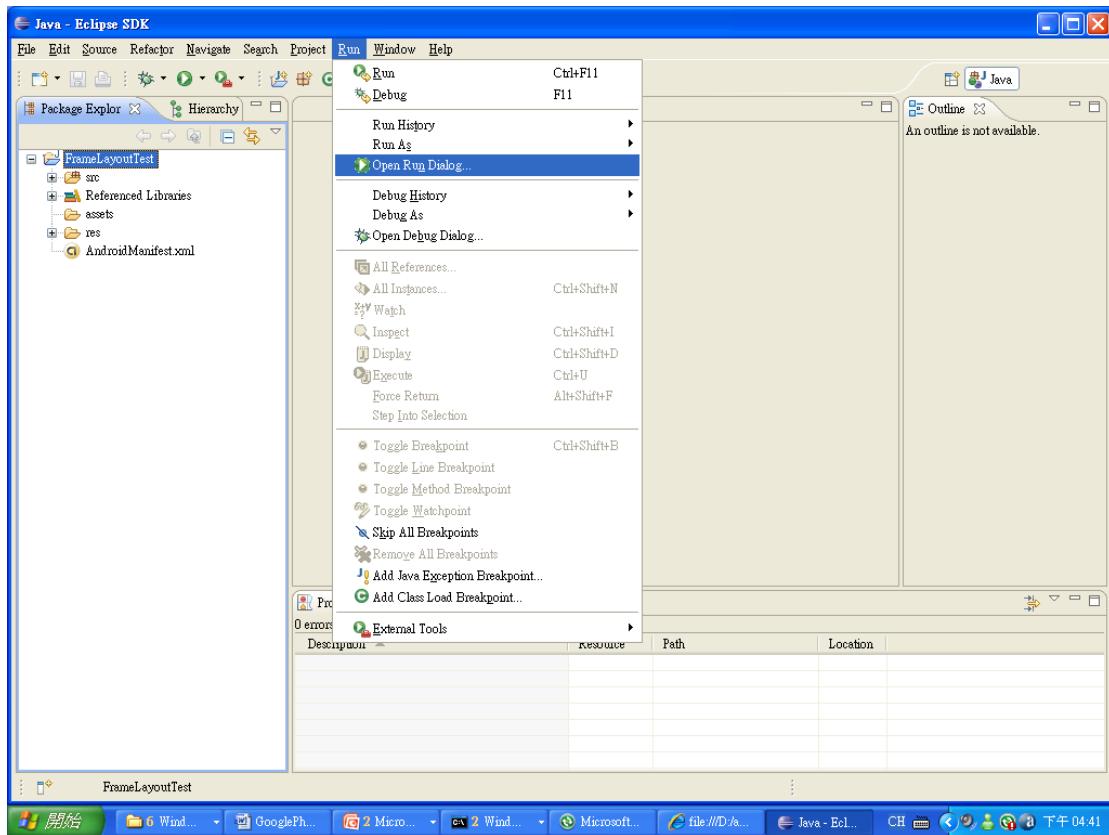
Select Window > Preferences... to open the Preferences panel. (Mac OS X: Eclipse > Preferences)

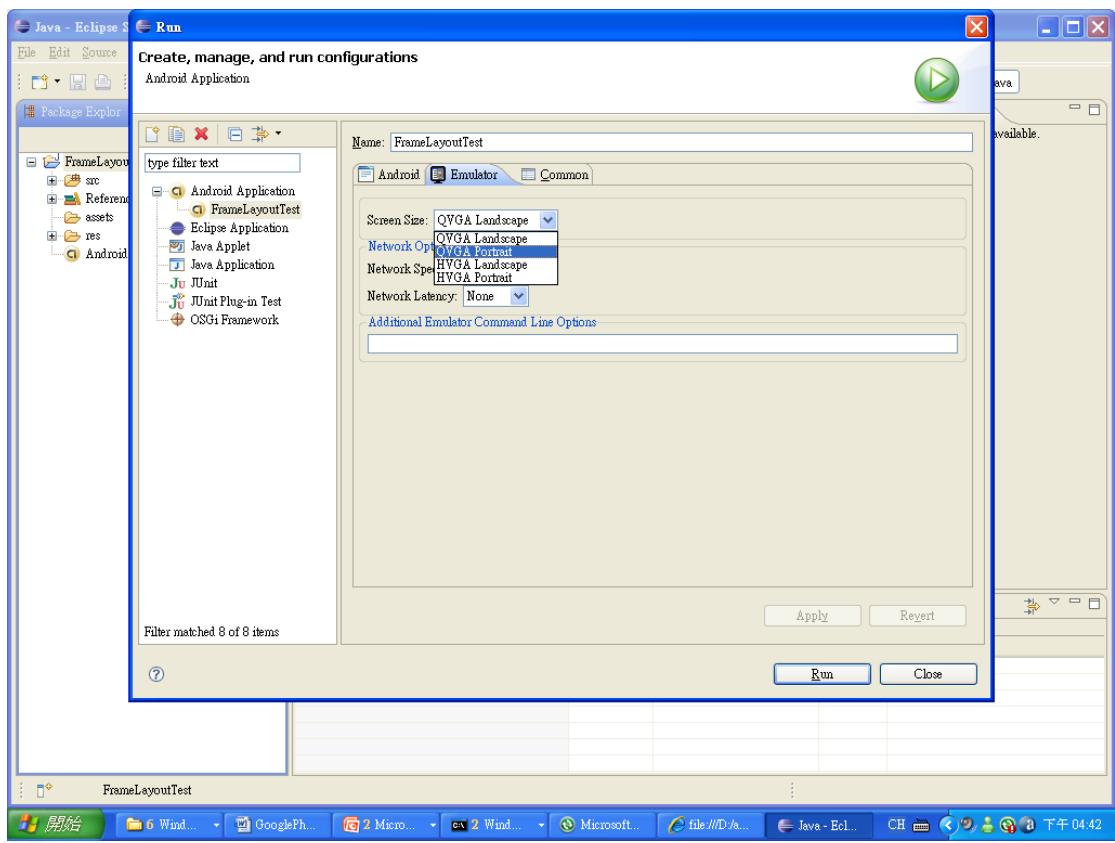
Select Android from the left panel.

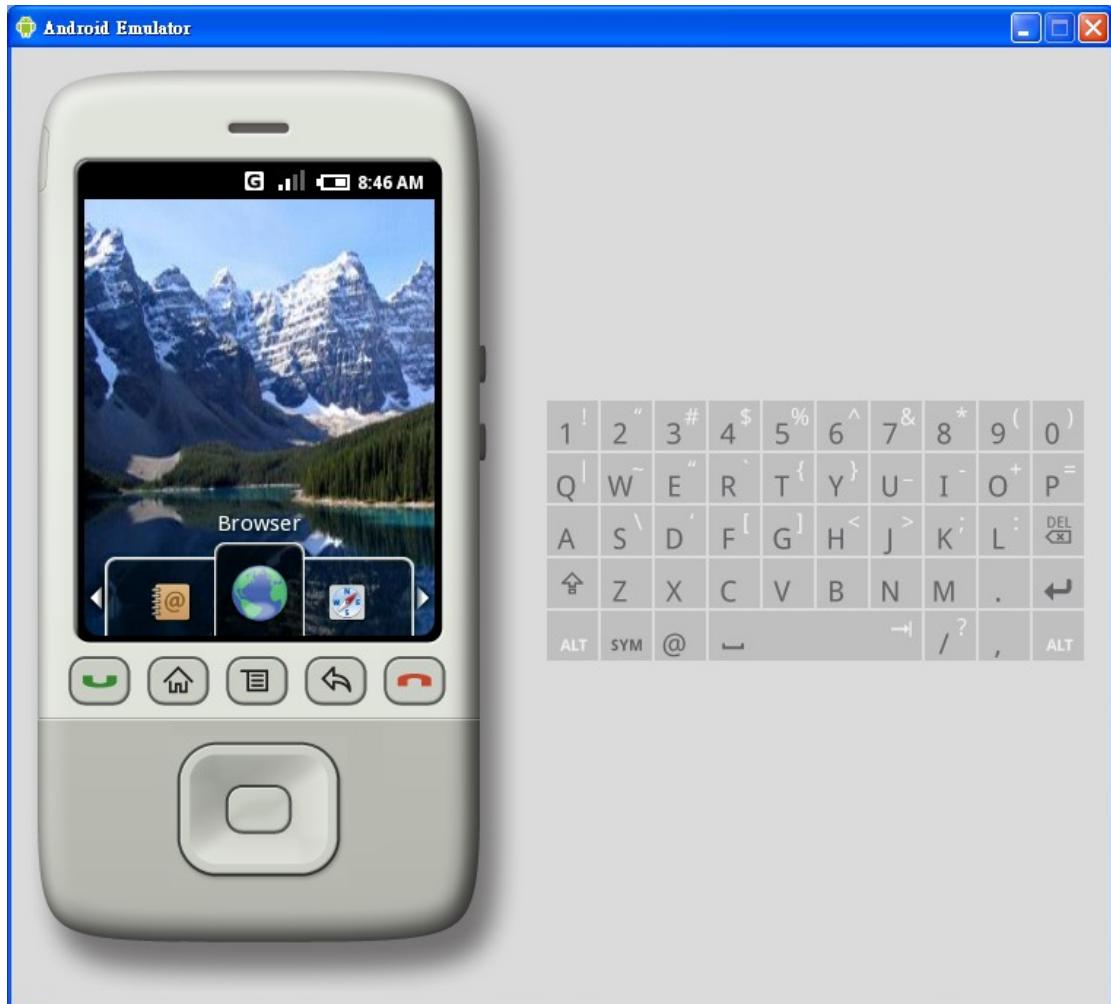
For the SDK Location in the main panel, press Browse... and find the SDK root directory.

Press Apply, then OK

更改執行環境





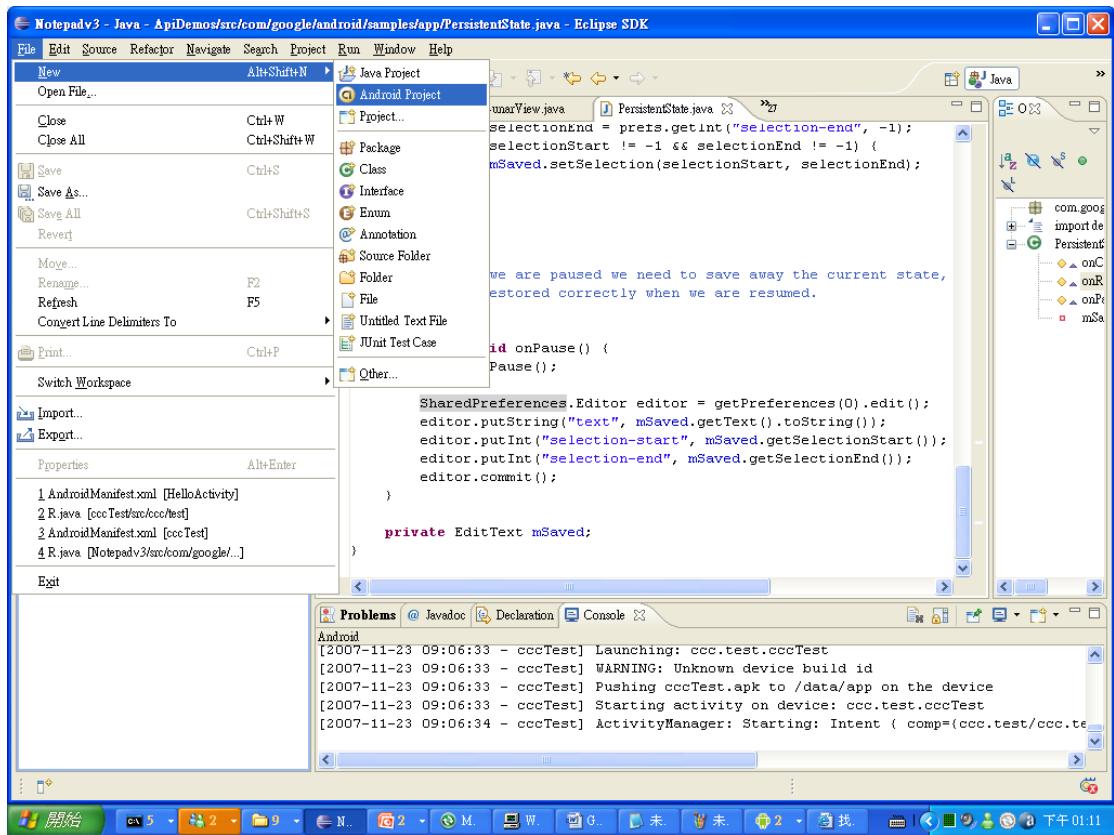


第3章 開發的第一個程式

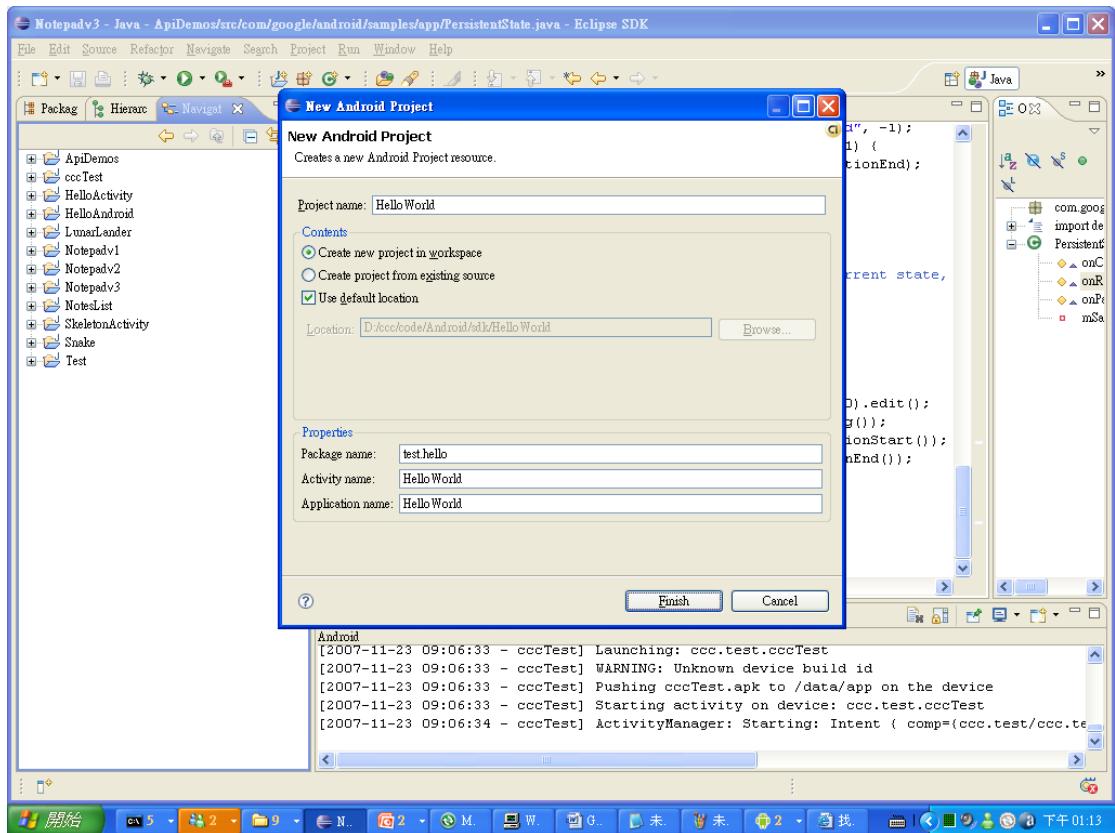
依照程式設計者的慣例，在撰寫第一個程式時，我們都會以最簡單的輸出 Hello World！訊息在銀幕上作為第一個程式，以下我們就遵循這樣一個慣例，在 Android 平台上開發出第一個屬於 Google Phone 的 Hello World！程式。

3.1 建立 Android 程式專案

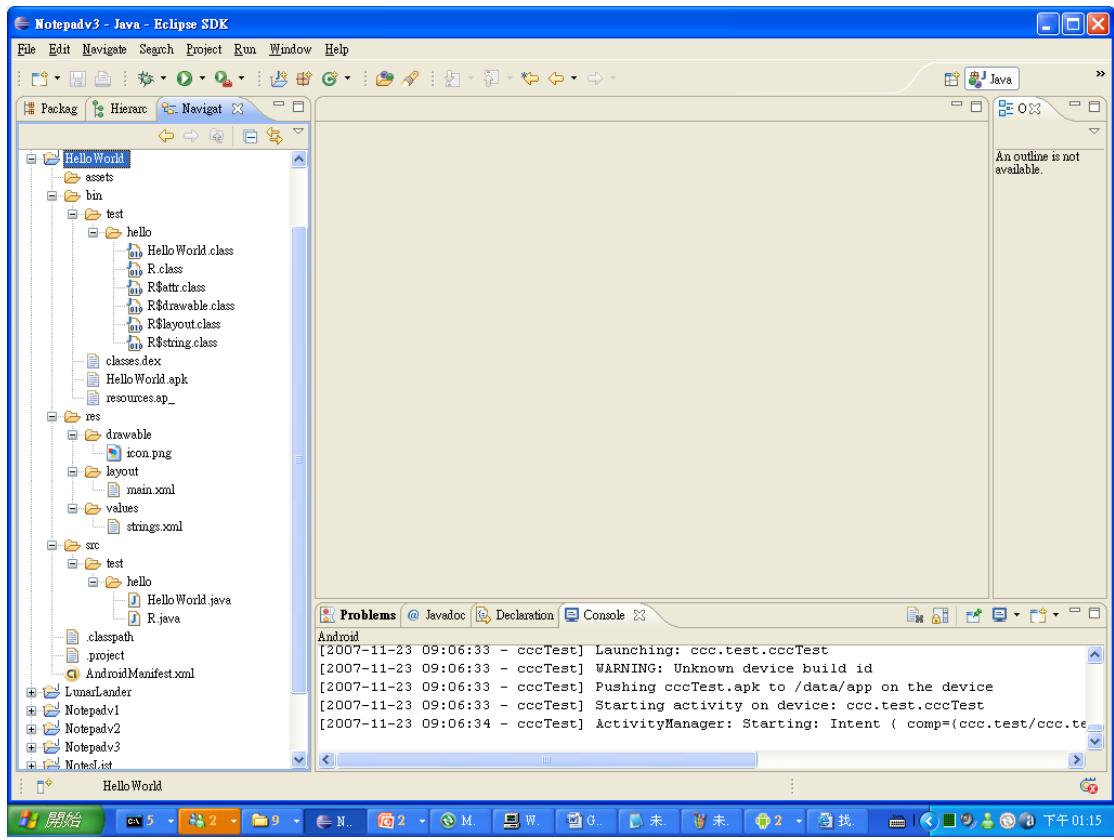
當你安裝好 Eclipse 之後，就可以開始撰寫程式了，首先要先建立一個專案，請點選 Eclipse 功能表中的 File/New/Android Project 的選項，如下圖所示：



當您點選後，會出現一個 New Android Project 的視窗，請在 Project Name 一欄中鍵入 HelloWorld 作為專案名稱，選擇 Create new project in workspace，並在 Package Name 中鍵入 test.hello 作為套件名稱，接著在 Activity Name 與 Application Name 欄位中都鍵入 HelloWorld 作為程式名稱，然後按下 Finish 鍵。

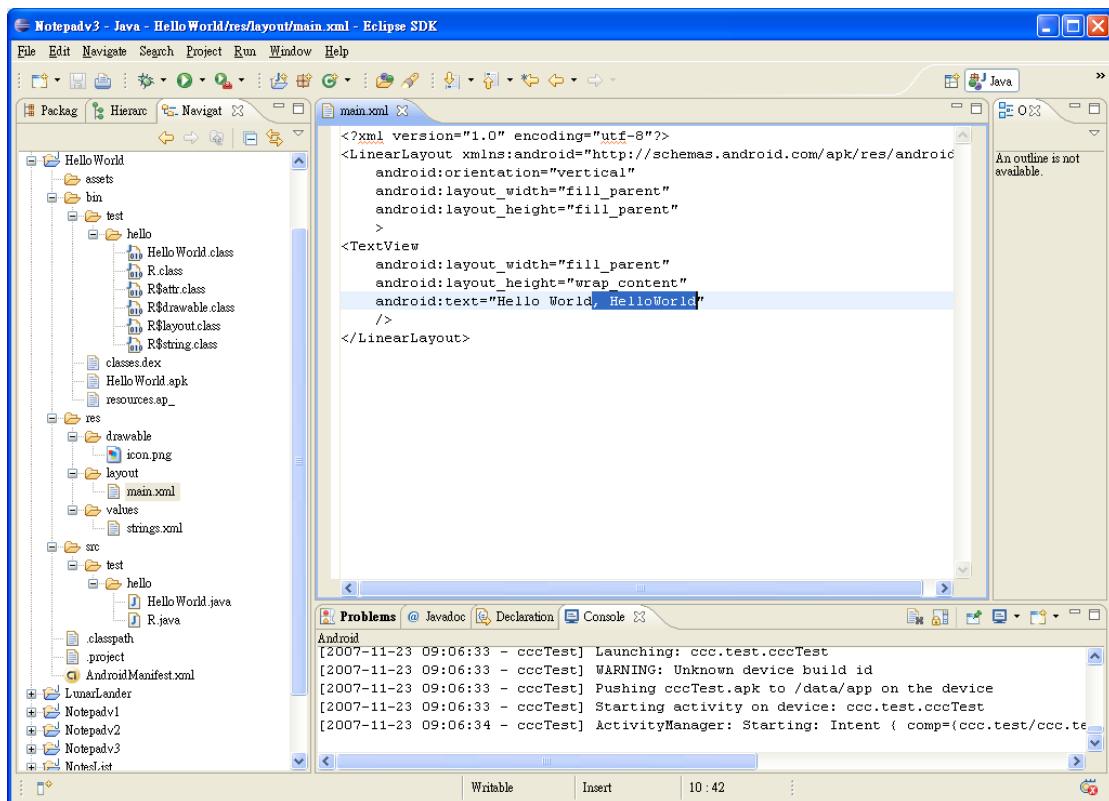


如此、Eclipse 就會建立一個名稱為 `HelloWorld` 的專案，並且自動產生很多檔案，其中、我們的主程式位在 `src/test/hello/HelloWorld.java` 中，另外、還有一個重要的檔案，是 `res/layout/main.xml` 檔。



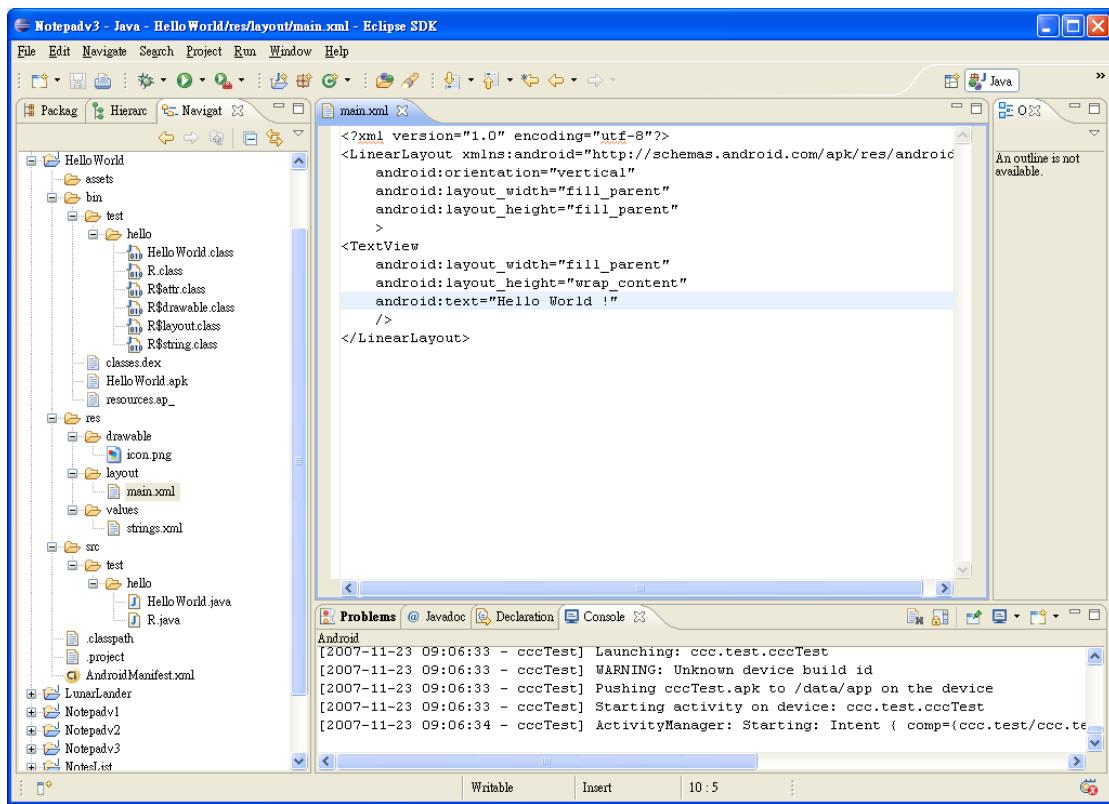
3.2 修改 XML 介面

請點選 res/layout/main.xml 檔案，您會看到如下圖的一個 XML 文件，請仔細觀察一下這個文件，猜猜看他的用途？



您可能已經猜到、Android 是使用 XML 來做視窗介面的描述 (微軟也正在做同樣的事，Microsoft.NET 3.5 與 Visual Studio 2008 正在建立一個稱為 Windows Presentation Foundation (WPF) 的顯示架構，同樣是使用 XML 來描述視窗介面)。

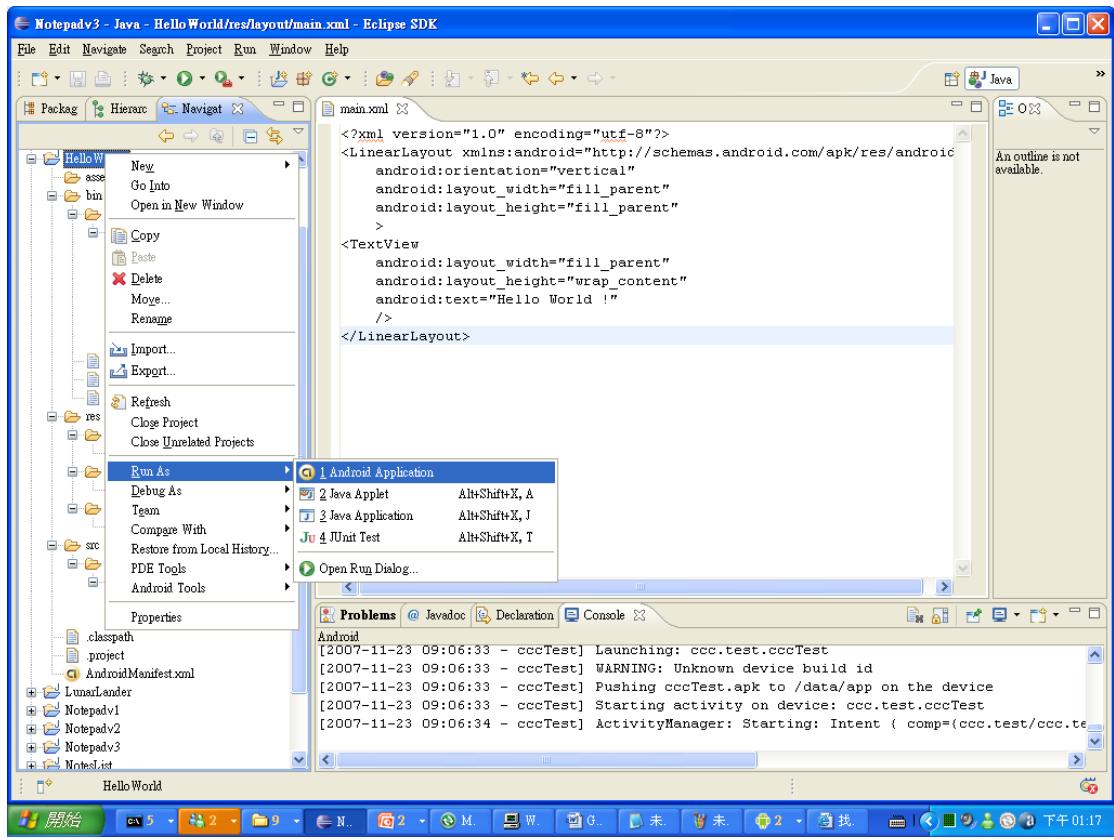
main.xml 中的最外層標記 **LinearLayout** 表示本畫面將使用線性排列的方式，其中的 **TextView** 代表在本畫面中會有一個文字區域，屬性 **android:text** 代表該文字區域中所顯示的文字，在此、請將其內容改為 **Hello World !**



如此、您便完成了第一個程式 **HelloWorld**。

3.3 執行專案程式

接著、您就可以在 Navigate 視窗中的 **HelloWorld** 專案上按下滑鼠右鍵，選擇 **Run As/Android Application** 功能，開始執行您的 **HelloWorld** 程式，如下圖所示：



稍後片刻之後，模擬器會被啓動，經過一小段開機程序，您會看到 **HelloWorld** 程式的執行結果畫面如下。



讀到這裡，可能很多人對 Android 的程式設計原理會有相當大的疑問，為了解開這些疑問，我們將更深入 HelloWorld 程式的其他檔案一窺究竟。

3.4 程式執行的原理

在 src/test/hello 資料夾中的 HelloWorld.java 程式如下所示

```
package test.hello;

import android.app.Activity;
import android.os.Bundle;

public class HelloWorld extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
    }
}
```

在上例中、**import** 指令引入了 `android.app.Activity` 與 `android.os.Bundle` 兩個類別，我們可以看到 `HelloWorld` 繼承了 `Activity` 這個類別，`Activity` 在 `Android` 平台當中的地位相當於一個作業系統的行程 (Process)，也就是相當於 `Java` 程式中的主程式 `main`，然而、在 `Android` 當中使用 `Activity` 物件來取代主程式的概念是有原因的，因為 `Android` 將主程式的概念也徹底的物件導向化了，於是屬於靜態函數的 `main` 搖身一變就成了物件版的 `Activity` 了。

為何 `Android` 要將主程式物件化呢，這牽涉到 `Android` 的設計理念，在 `Google Phone` 當中，程式一開始執行後就會一直存在，通常不會結束，這樣的做法在手機當中很自然，例如、我們正在玩遊戲的當中可能會有電話打進來，此時、我們會先去接個電話，等到電話講完了，我們又會回來繼續玩遊戲，因此、`Google Phone` 的程式必須要適應這種半途中斷的狀況，於是 `Android` 平台將主程式物件化，使得主程式永遠存在 `Google Phone` 中，然而、當越來越多的 `Activity` 被放入記憶體執行之後，卻可能使得手機的記憶體不堪負荷，而導致記憶空間不足的窘境，為了解決這樣的困擾，`Android` 的作業系統在必要時會將不重要的 `Activity` 先置換出去，等到該 `Activity` 又要被執行的時候才置換回來，於是 `Activity` 必須要在被置換出去前先儲存重要資訊，以便在被置換回來後得以收回這些資訊，以回復到先前的程式執行狀態，這樣也可節省行程

切換所產生的負擔，於是 Android 採用了物件化的 **Activity** 作為主程式的概念，使得 Android 中的程式變得很不一樣。

主程式被物件化為 **Activity** 之後，視覺化介面的作業也跟著要搭配 **Activity** 而設計，因而、Android 採用了 **View** 這一個概念，每個 **Activity** 都可以搭配一些 **View** 而被顯示出來，只要使用 **Activity** 中的 **setContentView** 函數即可，這也就是上述範例中 **setContentView(R.layout.main)** 指令的意義所在，當 **HelloWorld** 這個 **Activity** 被啓動後，立即將其顯示介面設為 **R.layout.main**，於是、**R.layout.main** 所對應到的介面將會被顯示出來。

然而、**R.layout.main** 到底對應到甚麼介面呢，這個答案很不明顯，但其實就是我們在 **res/layout/main.xml** 中所寫的那個介面。

當我們修改 **main.xml** 之後按下存檔按鈕時，Eclipse 會利用 Android 在 **tools** 資料夾下提供的 **Android Asset Packaging Tool (aapt)** 程式，立即產生 **R.layout.main** 以及對映的類別到 **bin/test/** 資料夾下的 **HelloWorld.apk** 壓縮檔中，這裏會包含 **R.layout.main** 所對應的物件，於是當我們使用 **setContentView(R.layout.main)** 指令的時候，**Activity** 就會指定該物件為其顯示物件。

第4章 Android 核心物件與程式架構

4.1 Android 的系統架構

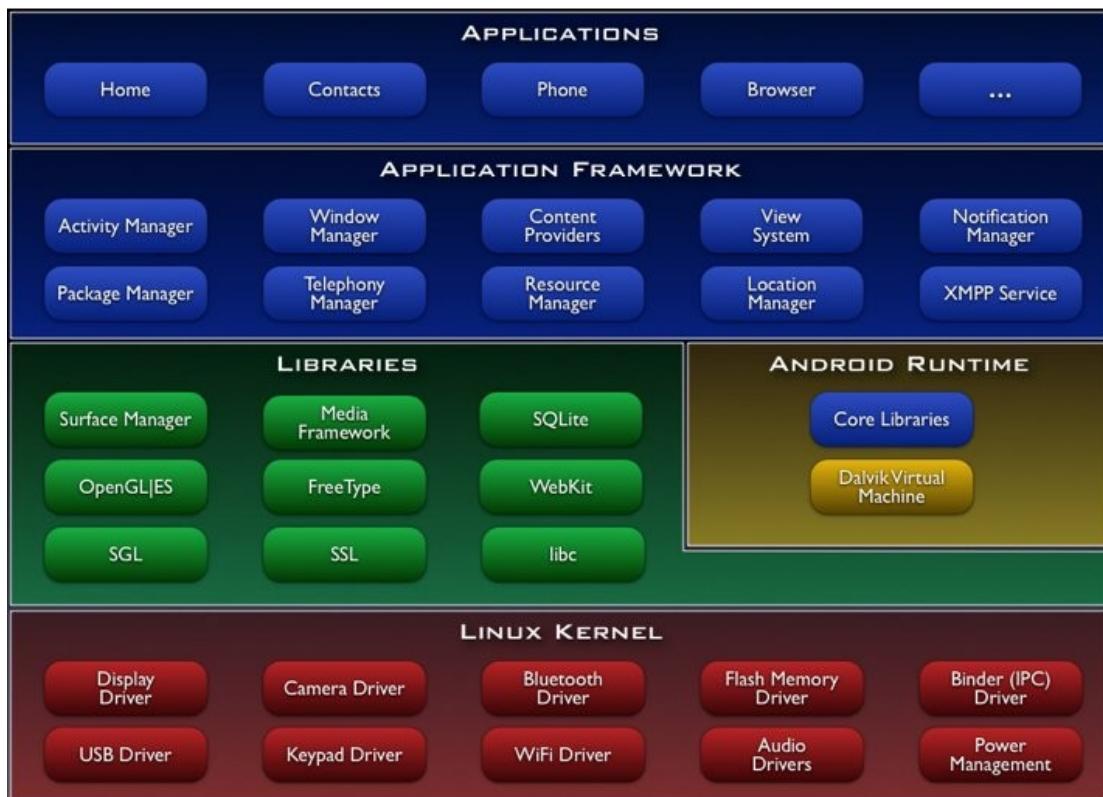


圖 4.1 Android 手機平台的架構

4.2 程式核心 - Activity 物件

4.3 重要的元件簡介

UI Elements and Concepts Glossary

Here is a list of common UI elements and concepts that you will see here and elsewhere in the SDK.

Activity

The standard screen in an Android application. Activity is a class that Android can start when a matching Intent is thrown by this or another application. Most commonly, it is visibly represented by a full screen window that can receive and handle UI events and perform complex tasks, because of the Window it uses to render its window. Though an Activity is typically full screen, it can also be floating or transparent.

View

A rectangular area on the screen that can be drawn to, handles click, keystroke, and other interaction events. A View is a base class for most components of an Activity or Dialog screen (text boxes, windows, and so on). It receives calls from its container object to draw itself, and informs its parent object about where and how big it would like to be (which may or may not be respected by the parent). It is represented by the base class View.

View Group

A container that holds multiple child View objects, deciding where they will be and how large they can be, and calling on them to draw themselves when appropriate. Some are invisible and for layout only, while others have a UI themselves (for instance, scrolling list boxes). View groups are all in the widget package, but extend ViewGroup.

Widget

A form element, such as a text box or popup menu. They have the ability to draw themselves and handle UI events. Widgets are all in the widget package.

Drawable

A visual element that is loaded into another UI element, typically as a background image. It does not receive events, but does assign various other properties such as "state" and scheduling to enable subclasses such as animation objects or image libraries. Many drawable objects are loaded from resource files — xml or bitmap files that describe the image. The base class is Drawable. See

Resources.

Panel

A panel is a concept not backed by a specific class. It is a View of some sort that is tied in closely to a parent window, but can handle clicks and perform simple functions related to its parent. A panel floats in front of its parent, and is positioned relative to it. A common example of a panel (implemented by Android) is the options menu available to every screen. At present, there are no specific classes or methods for creating a panel — it's more of a general idea.

Dialog

A dialog is a floating window that can have buttons, and acts as a lightweight form that is intended to, at most, perform a simple action (such as click a button) and perhaps return a value. It is not intended to persist in the history stack, contain complex layout, or perform complex actions. Android provides a default simple dialog for you with optional buttons, though you can define a dialog layout yourself. The base class is Dialog, and the helper methods to open a dialog box are the various Activity.showAlert() methods.

Window

An abstract class that specifies the elements of a generic window, such as the look and feel (title bar text, location and content of menus, and so on). Dialog and Activity use an implementation of this class to render a window. You should not need to implement this class.

Surface

A block of memory that gets composited to the screen. A Surface holds a Canvas object for drawing, and provides various helper methods to draw layers and resize the surface. You should not use this class directly; use SurfaceView instead.

SurfaceView

A View object that wraps a Surface for drawing, and exposes methods to specify its size and format dynamically. The camera app

uses SurfaceView for its preview screen. A SurfaceView provides a way to draw independently of the UI thread for resource-intense operations (such as games or camera previews), but it uses extra memory as a result. SurfaceView supports both Canvas and OpenGL ES graphics.

Canvas

A drawing surface where the actual bits are composited. It has methods for standard computer drawing of bitmaps, lines, circles, rectangles, text, and so on. It is bound to a Bitmap or Surface. Canvas is the simplest, easiest way to draw 2D objects on the screen. However, it does not support hardware acceleration, as OpenGL ES does.

OpenGL ES

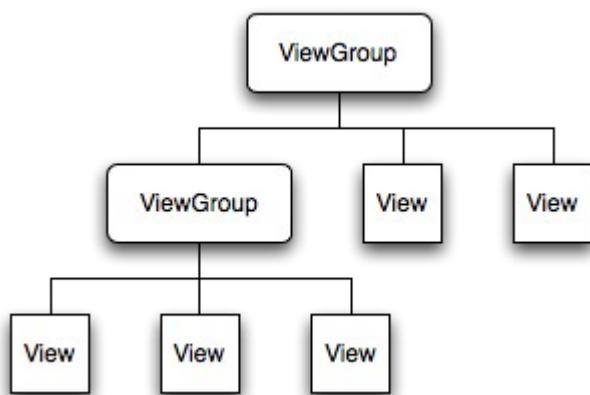
Android provides OpenGL ES libraries that you can use for fast, complex 3D images. It is much harder to use than a Canvas object, but better for 3D objects. The `graphics.gutils` package exposes OpenGL ES functionality.

第5章 視覺化介面的基礎

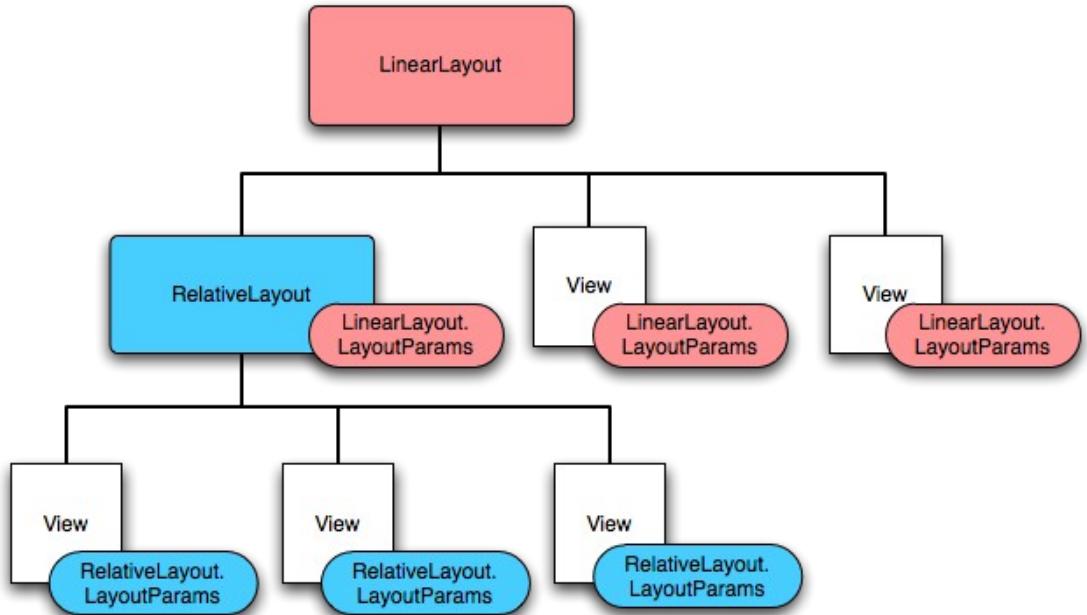
5.1 Android 介面設計簡介

在上一章我們曾經談到，Android 平台的視覺化介面是使用 XML 的方式設計的，然而上一章我們只做了一個簡單的 HelloWorld 的表單，在本章，我們將介紹如何利用常用的元件設計出較實用的表單，並展示每個常用視覺元件的用途。

Android 當中的視覺化界面元件，可分為『容器』與『非容器』兩類，容器類繼承 ViewGroup 物件，非容器類則從 View 衍生出來，如下圖所示。



這些 Layout 可以套疊式的組成一棵元件樹，其中、父節點的 Layout 與子節點的 LayoutParams 之間有控制關係，舉例而言、若父節點是 RelativeLayout，則子節點的元件中可以指定 RelativeLayout.LayoutParams 中的屬性，以控制子節點在父節點中的排列狀況，如下圖所示：



5.2 以 XML 設計視覺化介面

5.3 基本的視覺元件

5.4 各種排版元件 – Layout

容器類物件中專門用來排版者，稱為 **Layout**，以下是 Android 中 **Layout** 物件的列表：

元件名稱	說明
FrameLayout	單一物件的容器
AbsoluteLayout	以絕對座標排版的容器。
LinearLayout	線性(水平或垂直)排版的容器。
RelativeLayout	以相對座標(相對於父元件或兄弟元件)排版的容器。
TableLayout	以表格方式排版的容器。

5.4.1 FrameLayout

Android 中的 FrameLayout 與 Java Swing 中的 FrameLayout 完全不同，Android 中的 FrameLayout 代表只能放入單一視覺元件的容器，與 Swing 中區分『上、下、左、右、中』的 FrameLayout 完全是兩回事。





```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World, FrameLayoutTest"
    />
</FrameLayout>
```

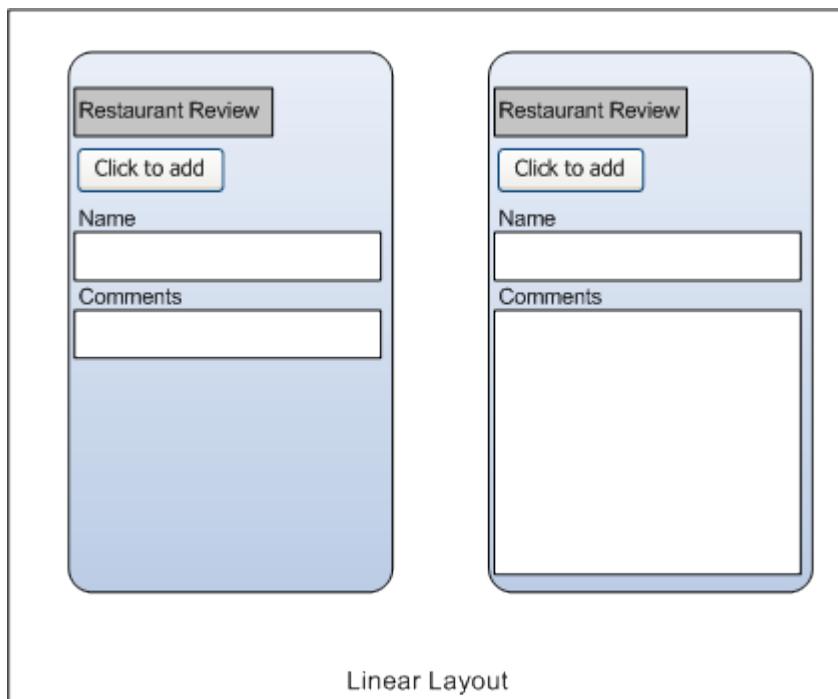
5.4.2 AbsoluteLayout



```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
```

```
<EditText id="@+id/title_edit"
    android:layout_x = "10px"
    android:layout_y = "10px"
    android:layout_width="200px"
    android:layout_height="wrap_content"/>
<Button id="@+id/button"
    android:layout_x = "220px"
    android:layout_y = "10px"
    android:text="Send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<TextView
    android:layout_x = "10px"
    android:layout_y = "50px"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World, AbsoluteLayoutTest"/>
</AbsoluteLayout>
```

5.4.3 LinearLayout



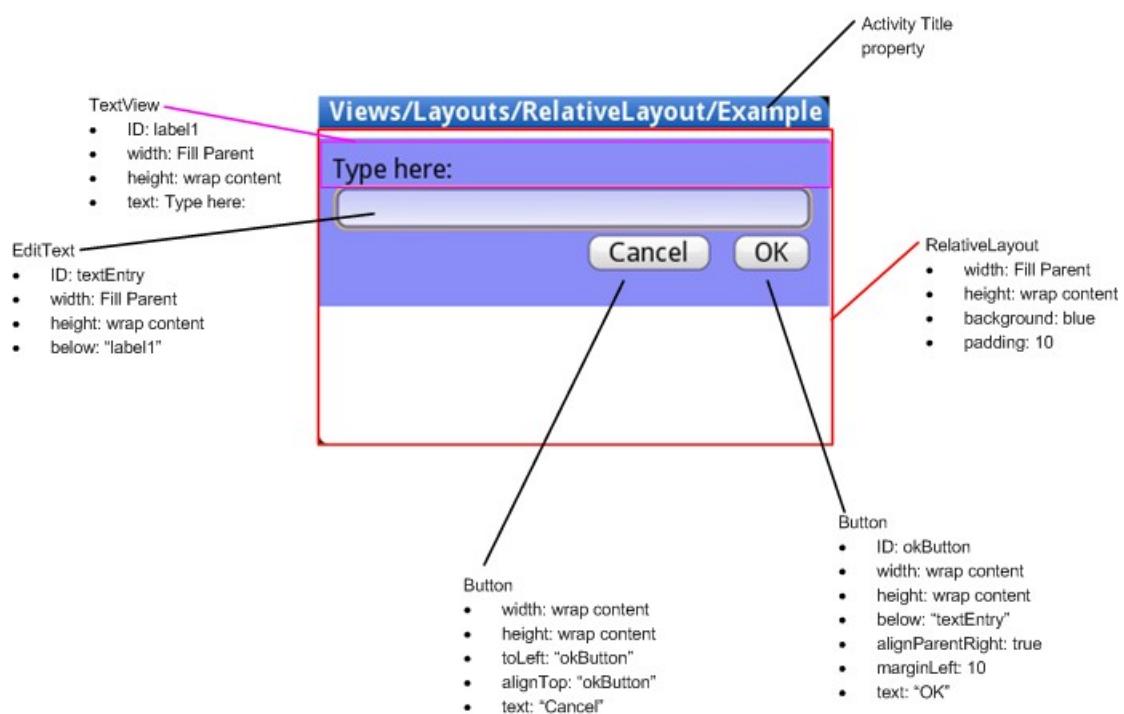
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
```

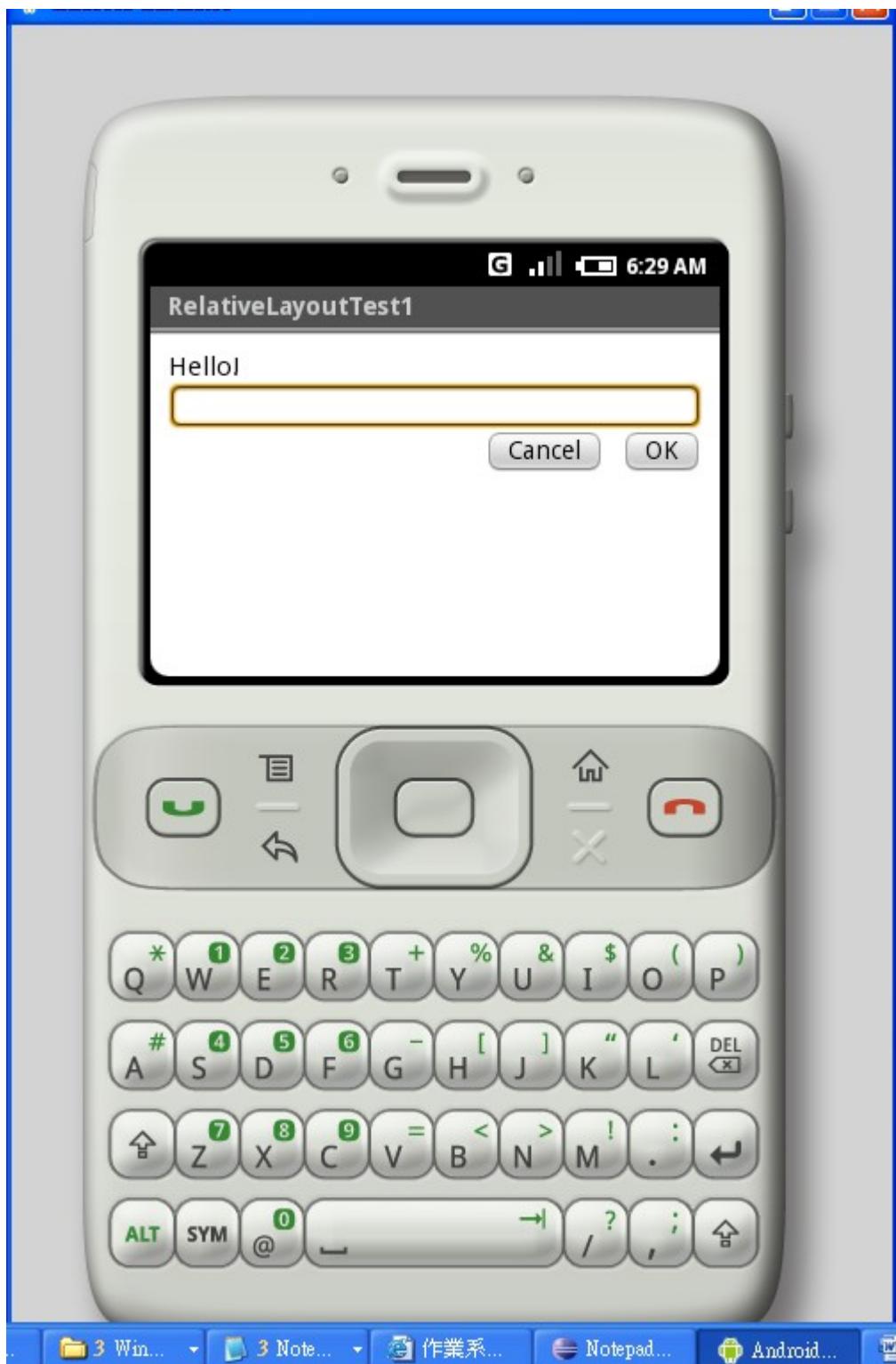
```

xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <EditText id="@+id/title_edit"
        android:layout_width="200px"
        android:layout_height="wrap_content"/>
    <Button id="@+id/button"
        android:text="Send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Hello World, AbsoluteLayoutTest"/>
</LinearLayout>

```

5.4.4 RelativeLayout





```
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:padding="10dip">
```

```
<TextView id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello!"/>

<EditText id="@+id/entry"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/label"/>

<Button id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="OK" />

<Button id="@+id/cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeft="@id/ok"
    android:layout_alignTop="@id/ok"
    android:text="Cancel" />
</RelativeLayout>
```

5.4.5 TableLayout



Table cell lines
(not actually
displayed in the
UI)

Table Layout



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    id="@+id/table_layout"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
```

```
<TableRow>
    <Button id="@+id/button1" android:text="Button1" />
    <Button id="@+id/button2" android:text="Button2" />
</TableRow>
<TableRow>
    <Button id="@+id/button3" android:text="Button3"
        android:layout_span="2"/>
</TableRow>
</TableLayout>
```

由於視覺化介面的物件屬性與 XML 標記的屬性並不是 1 對 1 對映的，因此、Android 的程式設計人原有必要學會看 Android SDK 當中的原始函式庫說明文件，文件中的說明大至按照傳統 Java 的格式，然而、其中與傳統 Java 最大的當屬 XML 屬性了，在附錄 D 中，我們以 **TableRow** 這個物件為例介紹其文件的編寫方式，以利讀者看懂這些說明文件，順利成為 Google Phone/Android 的高手。

第6章視覺化介面進階

6.1 視覺元件

取得銀幕大小

```
public void onCreate(Bundle icicle) {  
    ...  
    DisplayMetrics dm = new DisplayMetrics();  
    getWindowManager().getDefaultDisplay().getMetrics(dm);  
    String msg = "Display " + dm.widthPixels + " x " +  
    dm.heightPixels;  
    Log.i(LOG_ID, msg);
```

If you have an own View then override onLayout() and get the views dimension with getWidth() and getHeight().
and be prepared for the screen to change size dynamically.

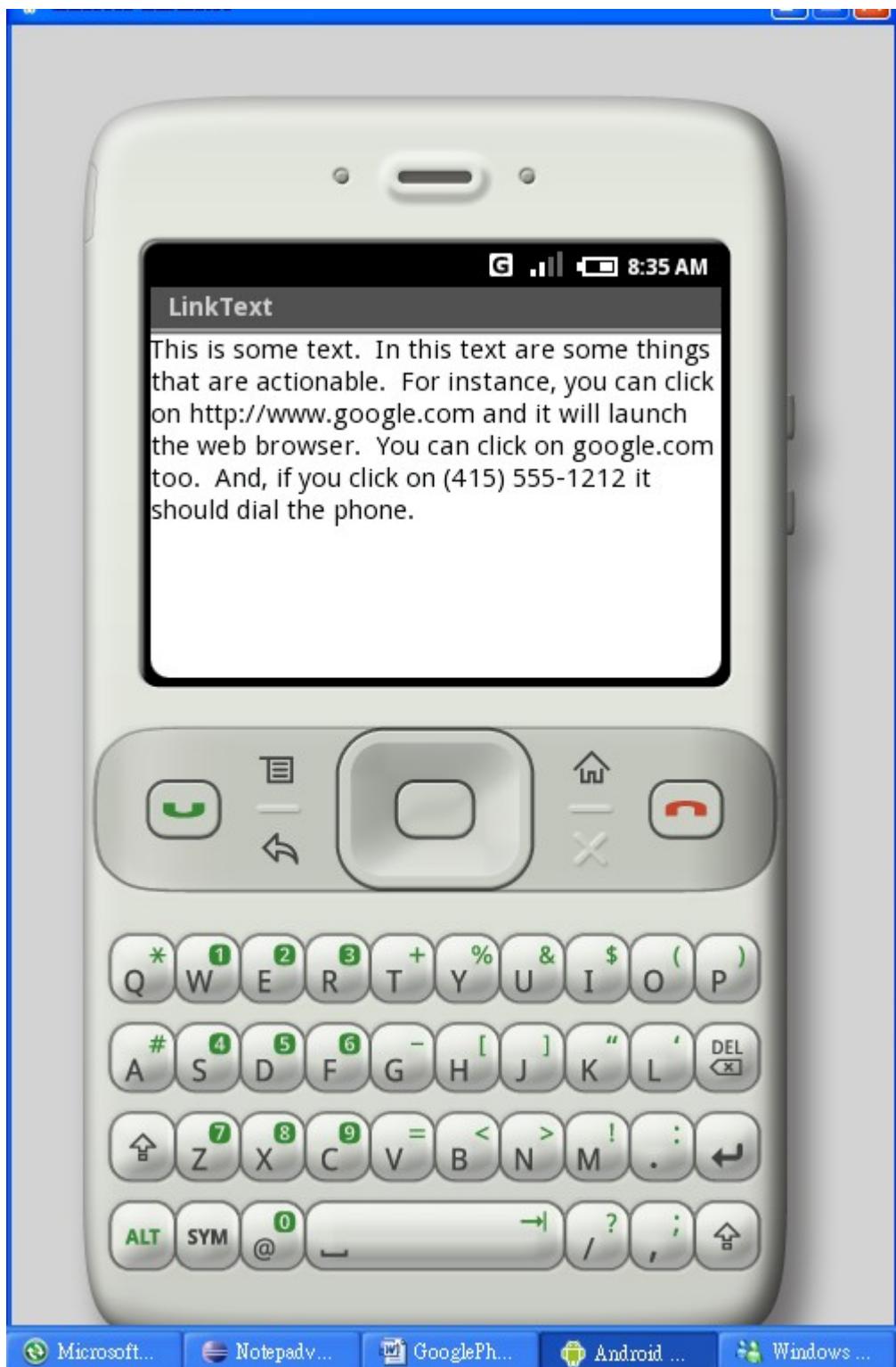
6.2 文字型元件

6.2.1 TextView

```
<?xml version="1.0" encoding="utf-8"?>  
<TextView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:autoLink="all"  
        android:text="This is some text. In this text are some things  
        that are actionable. For instance, you can click on  
        http://www.google.com and it will launch the web browser. You can  
        click on google.com too. And, if you click on (415) 555-1212 it  
        should dial the phone.">
```

/>

但是、若把 `android:autoLink="all"` 這行移除再執行，則會看到下列結果。



6.2.2 AutoCompleteTextView

```
package test.auto;
```

```
import test.auto.R;

import android.app.Activity;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.os.Bundle;

public class AutoCompleteTest extends Activity {

    @Override
    protected void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);

        ArrayAdapter<String> adapter = new
        ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, COUNTRIES);
        AutoCompleteTextView textView = (AutoCompleteTextView)
            findViewById(R.id.country_edit);
        textView.setAdapter(adapter);
    }

    static final String[] COUNTRIES = new String[] {
        "Albania", "Armenia", "Austria", "China", "Japan",
        "Korea", "Mexico", "Norway", "Singapore", "South Africa",
        "Taiwan", "Tajikistan", "Tanzania", "Thailand",
        "Turkey", "United Kingdom", "Zambia" };
}
```



6.3 圖片相關元件

Dialog

AlertDialog

DialogInterface
DatePickerDialog
ProgressDialog
TimePickerDialog

6.4 資料綁定元件

GallaryGrid
ViewListView
PopupList
ScrollView
Spinner
TabHost
ViewFlipper
ViewSwitcher

Gallery	A horizontal scrolling display of images, from a bound list.
GridView	Displays a scrolling grid of m columns and n rows.
ListView	Displays a scrolling single column list.
PopupList	A freestanding popup list of elements with a border.
ScrollView	A vertically scrolling column of elements.
Spinner	Displays a single item at a time from a bound list, inside a one-row textbox. These objects all hold child UI elements. Some provide visible UI, and others only handle child layout. Rather like a one-row listbox that can scroll either horizontally or vertically.
SurfaceView	Provides direct access to a dedicated drawing surface. It can hold child views layered on top of the surface, but is intended for applications that need to draw pixels, rather than using widgets.
TabHost	Provides a tab selection list that monitors clicks and enables the application to change the screen whenever a tab is clicked.
ViewFlipper	A list that displays one item at a time, inside a one-row textbox. It can be set to swap items at timed intervals, like a slide show.
ViewSwitcher	Same as ViewFlipper.

6.5 XML 屬性的設定

6.6 統一的風格元件

第 1 節 在整個專案中使用統一的風格 (Theme)

Applying a Theme to your Application

If you do not explicitly specify a theme for your UI, Android will use the default theme defined by android.R.style.Theme. Many times you will want to use a different system theme (such as Theme.Dark) or create your own theme (as described in Style and Theme Resources).

To set your theme in XML, simply specify the desired theme in your `AndroidManifest.xml` file with the `theme` attribute. This can be used with the `<application>` tag (shown here) to specify a default theme for all of your activities, and/or with the `<activity>` to control the theme of a particular activity.

第7章事件處理

7.1 事件處理的基本概念

Android 所使用的事件處理方式，在視覺化介面上，是以傳統 Java 的 Listener 的方式，覆寫 Override 父類別中對映的事件函數的方式處理的，這和 VB 等設計方式很類似，在 7.2 節我們將使用一個簡單的按鈕控制示範這種覆寫處理法。

然而、在自訂的事件方面，Android 是以 訊息傳遞的方式，搭配控制作業系統排程的方式處理的，這是一個相當奇特的處理方式，也使得初學者很難理解這種概念，在 7.3 節中、我們將使用此種方式，設計一個小的電子鐘，以說明此種訊息傳遞的事件處理方式。

7.2 按鈕事件

```
public class SendResult extends Activity
{
    /**
     * Initialization of the Screen after it is first created.
     * Must at least call setContentView() to
     * describe what is to be displayed in the screen.
     */
    protected void onCreate(Bundle savedInstanceState)
    {
        ...
        // Listen for button clicks.
        Button button = (Button) findViewById(R.id.corky);
        button.setOnClickListener(mCorkyListener);
    }
}
```

```
// Create an anonymous class to act as a button click listener.  
private OnClickListener mCorkyListener = new OnClickListener()  
{  
    public void onClick(View v)  
    {  
        // To send a result, simply call setResult() before  
your  
        // activity is finished.  
        setResult(RESULT_OK, "Corky!");  
        finish();  
    }  
}  
}
```

7.3 滑鼠事件

7.4 鍵盤事件

第 1 節 在程式中取得元件

取得上層元件

使用 R.id..... 可取得 layout 中用 XML 所定義的視覺化元件。

取得內層元件

使用 Activity.findViewById 可取得內層的元件，

```
TextView msgTextView = (TextView) findViewById(R.id.msg);  
msgTextView.setText(R.string.push_me);
```

第 2 節 撰寫事件驅動函數

Listening for UI Notifications

Some UI notifications are automatically exposed and called by
Android. For instance, Activity exposes overrideable methods

onKeyDown and onKeyUp, and Widget exposes onFocusChanged(boolean, int). However, some important callbacks, such as button clicks, are not exposed natively, and must be registered for manually, as shown here

```
public class SendResult extends Activity
{
    /**
     * Initialization of the Screen after it is first created. Must at least
     * call setContentView() to
     * describe what is to be displayed in the screen.
     */
    protected void onCreate(Bundle savedValues)
    {
        ...

        // Listen for button clicks.
        Button button = (Button)findViewByld(R.id.corky);
        button.setOnClickListener(mCorkyListener);
    }

    // Create an anonymous class to act as a button click listener.
    private OnClickListener mCorkyListener = new OnClickListener()
    {
        public void onClick(View v)
        {
            // To send a result, simply call setResult() before your
            // activity is finished.
            setResult(RESULT_OK, "Corky!");
            finish();
        }
    };
}
```

第 3 節 一個簡單的按鈕範例





```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
>
<Button id="@+id/buttonHello"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Press me to say hello !"
/>
</LinearLayout>

package test.widget;

import test.widget.R;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class ButtonTest extends Activity {
    Activity me;
    Button buttonHello;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
        me = this;
        buttonHello = (Button) this.findViewById(R.id.buttonHello);
        buttonHello.setOnClickListener(buttonHelloListener);
    }
    private View.OnClickListener buttonHelloListener
    = new View.OnClickListener() {
        public void onClick(View arg0) {
            me.showAlert("Title", "Hello!", "OK", true);
        }
    };
}
```

7.5 時間事件



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1.0" />
    <TextView id="@+id/timeTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1.0"
        android:layout_gravity="center_horizontal"
        android:text="" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1.0" />
</LinearLayout>
```

```
package test.clock;

import test.clock.R;
import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.SystemClock;
import android.view.View;
import android.widget.TextView;
import java.util.Calendar;

public class DigitalClock extends Activity {
```

```
View view;
TextView timeTextView = null;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);
    view = this.getCurrentFocus();
    timeTextView = (TextView) this.findViewById(R.id.timeTextView);
    timeTextView.setTextSize(48);
    Message msg = mHandler.obtainMessage(CLOCK);
    mNextTime = SystemClock.uptimeMillis();
    mHandler.sendMessageAtTime(msg, mNextTime);
}

private static final int CLOCK = 1;
private long mNextTime;

private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        if (msg.what == CLOCK) {
            long current = SystemClock.uptimeMillis();
            mNextTime = current + 1000;
            msg = obtainMessage(CLOCK);
            sendMessageAtTime(msg, mNextTime);
            String timeStr = String.format("%tT",
                Calendar.getInstance());
            timeTextView.setText(timeStr);
        }
    }
};
```

7.6 Notification 通知事件的處理

第8章 資源的使用

8.1 Android 中的資源概念

8.2 Simple Values

Colors

Strings

Dimensions

8.3 Drawables

8.4 Animation

8.5 Layout

8.6 Style

8.7

8.8

第9章系統程式相關物件

9.1 Android 的系統程式簡介

9.2 程式核心 - Activity 物件

9.3 Bundle

9.4 Message

9.5 連接者 - Intent 物件

9.6 服務 - Service 物件

9.7 資料提供者 - Provider 物件

9.8 AlarmManager

```
import android.app.ListActivity;
import android.content.Intent;
import android.content.PackageManager;
import android.content.PackageManager.ResolveInfo;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.SimpleAdapter;
```

第10章字串處理與 XML

自從全球資訊網 Web 被建立以後，文字型的資料處理就一直是程式設計師所必需具備的基本技巧，目前最常見的三種文字處理技術是 1. 字串物件 2. Regular Expression 比對 3. XML 文件處理。

10.1 Android 中的字串相關函式庫

java.lang

java.text

android.text Provides classes used to render or track text and text spans on the screen.

android.text.method Provides classes that monitor or modify keypad input.

android.text.style Provides classes used to view or change the style of a span of text in a View object.

android.text.util

10.2 字串物件的使用

10.3 正規表示式

java.util.regex

10.4 處理 XML 文件

android.sax

javax.xml.parsers

org.json

org.w3c.dom

org.xml.sax This package provides the core SAX APIs.

org.xml.sax.ext This package contains interfaces to SAX2 facilities that conformant SAX drivers won't necessarily support.

org.xml.sax.helpers This package contains "helper" classes, including support for bootstrapping SAX-based applications.

第11章 資料儲存

11.1 共用資料 – SharedPreference

第1節 SharedPreferences

Preference 是 Android 中用來儲存資料最簡單的一種方法，我們可以使用：
Context.getSharedPreferences() 或 Activity.getPreferences() 取得儲存的變數，或對某一 Preference 變數的 editor 物件進行 put 動作以儲存資料，但這些共用資料只能在同一個 Package 底下使用。

以下是使用 Preference 來設定一個小計算機中按鍵模式 keypress 的範例程式：

```
public class Calc extends Activity {  
    public static final String PREFS_NAME = "MyPrefsFile";  
  
    ...  
  
    @Override  
    protected void onCreate(Bundle state){  
        super.onCreate(state);  
  
        ...  
  
        // Restore preferences  
        SharedPreferences settings =  
        getSharedPreferences(PREFS_NAME, 0);  
        boolean silent = settings.getBoolean("silentMode", false);  
        setSilent(silent);  
    }  
  
    @Override  
    protected void onStop(){  
        super.onStop();
```

```
// Save user preferences. We need an Editor object to  
// make changes. All objects are from android.context.Context  
SharedPreferences settings =  
getSharedPreferences(PREFS_NAME, 0);  
SharedPreferences.Editor editor = settings.edit();  
editor.putBoolean("silentMode", mSilentMode);  
  
// Don't forget to commit your edits!!!  
editor.commit();  
}  
}
```

11.2 檔案系統 - Files

第2節 Files

Context.openFileOutput() and Context.openFileInput()

If you have static files to package with your application at compile time, you can save your file in your project in res/raw/<mydatafile>, and then get it with Resources.openRawResource (R.raw.mydatafile).

To understand why we cannot use standard java file access, like:

Java:

```
FileWriter f = new FileWriter("impossible.txt");  
// throws: 'java.io.FileNotFoundException: /impossible.txt '
```

, we have to understand the Security-Model of Android.

Each *.apk File that is installed on the Emulator/Device gets its own User-ID from the Linux System. This ID is the key to the sandbox of the application. This 'sandbox' protects the application (and its files) from other bad Twisted Evil apps, that i.e. want to manipulate the files we created in a bad manner (Like writing into them: "Whos reads this is... dumb ! Hahaha").

Note wrote:

But, writing to SD-Cards is still possible with 'normal' Java methods
Exclamation

Java:

```
FileWriter f = new FileWriter("/sdcard/download/impossible.txt");
```

will work without any problem, you just need to add virtual sdcard to your emulator.

11.3 資料庫 - Sqlite

第 3 節 Sqlite

To create a database, use Context.createDatabase() and Context.openDatabase(), and read and write this data as appropriate

11.4 資料提供者 - ContentProvider

第 4 節 Content Provider

Android 中要跨越應用存取資料，Content Provider 是唯一的方法，Content Provider 是可以被所有應用存取得一種物件。

第12章網路函式庫

Android 的網路函式庫採用了標準的 `java.net.*` 中的網路函式庫，包含 `Socket`, `URL`, `InetAddress`, 等物件，並且將 Android 所擴充的物件放在 `android.net.*` 中，以補充原先 `java` 網路函數上的不足，另外、Android 也從 apache 當中取用了 `httpClient` 的函式庫以形成更完整的網路架構。

12.1 網路相關函式庫

12.2 Socket 程式設計

12.3 Apache 的 HttpClient

12.4 Android 的網路函式庫

12.5 XMPP 網路服務協定

XMPP : ineXtensible Messaging and Presence Protocol

Totally *Unofficial* Android GTalk Client (Send/Receive XMPP Messages)

<http://davanum.wordpress.com/2007/11/23/totally-unofficial-android-gtalk-client-sendreceive-xmpp-messages/>



```
package org.apache.gtalk;
```

```
import android.app.Activity;
```

```
import android.app.NotificationManager;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.database.Cursor;
import android.os.Bundle;
import android.os.DeadObjectException;
import android.os.IBinder;
import android.provider.Im;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.*;
import com.google.android.xmppService.IXmppService;
import com.google.android.xmppService.IXmppSession;
import com.google.android.xmppService.Presence;

public class GTalkClient extends Activity implements
View.OnClickListener {
    private static final String LOG_TAG = "GTalkClient";

    IXmppSession mXmppSession = null;
    EditText mSendText;
    ListView mListMessages;
    EditText mRecipient;
    Button mSend;
    Button mSetup;

    /**
     * Called with the activity is first created.
     */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
```

```
// gather the troops
mSendText = (EditText) findViewById(R.id.sendText);
 mListMessages = (ListView) findViewById(R.id.listMessages);
mRecipient = (EditText) findViewById(R.id.recipient);
mSend = (Button) findViewById(R.id.send);
mSetup = (Button) findViewById(R.id.setup);

// set up handler for on click
mSetup.setOnClickListener(this);
mSend.setOnClickListener(this);

bindService((new Intent()).setComponent(
    com.google.android.xmppService.XmppConstants.XMPP_SERVICE_COMPONENT),
    null, mConnection, 0);
}

/**
 * Let the user know there was an issue
 *
 * @param msg
 */
private void logMessage(CharSequence msg) {
    NotificationManager nm = (NotificationManager)
getSystemService(
        Context.NOTIFICATION_SERVICE);

    nm.notifyWithText(123, msg, NotificationManager.LENGTH_LONG,
null);
}

/**
 * Setup the XMPP Session using a service connection
 */
private ServiceConnection mConnection = new ServiceConnection() {
```

```
    public void onServiceConnected(ComponentName className,
IBinder service) {
    // This is called when the connection with the
XmppService has been
    // established, giving us the service object we can use
to
    // interact with the service.  We are communicating with
our
    // service through an IDL interface, so get a client-side
    // representation of that from the raw service object.
    IXmppService xmppService =
IXmppService.Stub.asInterface(service);

    try {
        mXmppSession = xmppService.getDefaultSession();
        if (mXmppSession == null) {
            // this should not happen.
            logMessage(getText(R.string.xmpp_session_not_foun
d));
        }
        mXmppSession.setPresence(new
Presence(Im.PresenceColumns.AVAILABLE, "Am here now!"));
        } catch (DeadObjectException ex) {
        Log.e(LOG_TAG, "caught " + ex);
        logMessage(getText(R.string.found_stale_xmpp_service)
);
    }

    mSendText.setEnabled(true);
}

public void onServiceDisconnected(ComponentName
componentName) {
    // This is called when the connection with the service
has been
```

```
// unexpectedly disconnected -- that is, its process
crashed.

mXmppSession = null;
mSendText.setEnabled(false);
}

};

/***
 * Handle clicks on the 2 buttons
 *
 * @param view
 */
public void onClick(View view) {
    if (view == mSetup) {
        Log.i(LOG_TAG, "onClick - Setup");
        // Run a query against CONTENT_URI =
"content://im/messages"

        Cursor cursor = managedQuery(Im.Messages.CONTENT_URI,
null,
                "contact='"
                + mRecipient.getText().toString()
                +
                "'", null, null);
        // Display the cursor results in a simple list
        // Note that the adapter is dynamic (picks up new entries
        automatically)
        ListAdapter adapter = new SimpleCursorAdapter(this,
                android.R.layout.simple_list_item_1,
                cursor, // Give the cursor to the list adapter
                new String[]{Im.MessagesColumns.BODY},
                new int[]{android.R.id.text1});

        this.mListMessages.setAdapter(adapter);
    } else if (view == mSend) {
        // use XmppService to send data message to someone
        String username = mRecipient.getText().toString();
        if (!isValidUsername(username)) {
```

```
        logMessage(getText(R.string.invalid_username));
        return;
    }

    if (mXmppSession == null) {
        logMessage(getText(R.string.xmpp_service_not_connecte
d));
        return;
    }

    try {
        mXmppSession.sendTextMessage(username, 0,
mSendText.getText().toString());
    } catch (DeadObjectException ex) {
        Log.e(LOG_TAG, "caught " + ex);
        logMessage(getText(R.string.found_stale_xmpp_service)
);
        mXmppSession = null;
    }
}

private boolean isValidUsername(String username) {
    return !TextUtils.isEmpty(username) &&
username.indexOf('@') != -1;
}
```

12.6 WebKit 瀏覽器的控制

Android 中所採用的瀏覽器是 WebKit，和 iPhone 中的一樣，要控置瀏覽器很簡單，只要 import android.webkit.WebView 後，呼叫 loadUrl() 函數載入網頁即可。

```
package test.web;

import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;

public class WebViewTest extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        WebView webView = new WebView(this);
        webView.loadUrl("http://www.google.com/pda/");
        setContentView(webView);

    }
}
```





第13章二維繪圖功能

Android 中的二維繪圖功能主要在 `android.graphics` 函式庫中，重要的物件有 `Bitmap`, `Canvas`, `Paint` 等等，

13.1 Android 的繪圖相關物件

`android.graphics.Camera`

`android.graphics.drawable.Drawable`

`Paint`

`setStyle`

`setColor`

`setStrokeWidth(strokeWidth);`

`setAntiAlias(true);`

`setARGB(255, 255, 255, 255);`

`Path`

`PathEffect`

`Matrix`

`.setPolyToPoly(src, 0, dst, 0, src.length >> 1);`

`Paint.FontMetrics`

`mFontMetrics = mPaint.getFontMetrics();`

13.2 繪製基本圖形

13.3 整合練習 - 繪圖版範例



```
package test.paint;

import android.app.Activity;
import android.content.Context;
```

```
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;

public class PaintActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(new PaintView(this));
    }
}

class PaintView extends View {
    Bitmap mBitmap;
    Canvas mCanvas;
    private final Paint mPaint;
    private int strokeWidth = 2;

    public PaintView(Context c) {
        super(c);
        mPaint = new Paint();
        mPaint.setStrokeWidth(strokeWidth);
        mPaint.setAntiAlias(true);
        mPaint.setARGB(255, 255, 255, 255);
    }

    @Override
    protected void onSizeChanged(int w, int h, int oldw, int oldh)
    {
        int curW = mBitmap != null ? mBitmap.width() : 0;
        int curH = mBitmap != null ? mBitmap.height() : 0;
        if (curW >= w && curH >= h) {
```

```
        return;
    }

    if (curW < w) curW = w;
    if (curH < h) curH = h;

    Bitmap newBitmap = Bitmap.createBitmap(curW, curH, false);
    Canvas newCanvas = new Canvas();
    newCanvas.setDevice(newBitmap);

    if (mBitmap != null) {
        newCanvas.drawBitmap(mBitmap, 0, 0, null);
    }

    mBitmap = newBitmap;
    mCanvas = newCanvas;
}

@Override
protected void onDraw(Canvas canvas) {
    if (mBitmap != null) {
        canvas.drawBitmap(mBitmap, 0, 0, null);
    }
}

private int mCurX, mCurY, mLastX, mLastY;
boolean isDown = false;
@Override
public boolean onMotionEvent(MotionEvent event) {
    mCurX = (int)event.getX();
    mCurY = (int)event.getY();

    int action = event.getAction();
    if (action == MotionEvent.ACTION_DOWN) {
        isDown = true;
        mLastX = mCurX;
        mLastY = mCurY;
    }

    if (action == MotionEvent.ACTION_UP) {
```

```
        isDown = false;
    }

    if (isDown && action == MotionEvent.ACTION_MOVE) {
        if (mBitmap != null) {
            mCanvas.drawLine(mLastX, mLastY, mCurX, mCurY,
mPaint);
            invalidate();
        }
        mLastX = mCurX;
        mLastY = mCurY;
    }
    return true;
}
}
```

第14章2D 動畫功能

14.1

Bitmap

```
createBitmap(int[] colors, int width, int height, boolean hasAlpha)
int    getPixel(int x, int y)
int    height()
void  setPixels(int[] pixels, int offset, int stride, int x, int y, int
width, int height)
int    width()
```

Canvas :

```
setDevice()
boolean clipPath(Path path, ClipMode mode)
boolean clipRect(float left, float top, float right, float bottom)
    Intersect the current clip with the specified rectangle, which is
    expressed in local coordinates.
void   drawBitmap(Bitmap bitmap, Rect src, Rect dst, Paint paint)
void   drawBitmap(Bitmap bitmap, float left, float top, Paint paint)
void   drawBitmap(Bitmap bitmap, Rect src, RectF dst, Paint paint)
void   drawCircle(float cx, float cy, float radius, Paint paint)
void   drawColor(int color)
void   drawColor(int color, Mode mode)
void   drawLine(float startX, float startY, float stopX, float stopY,
Paint paint)
void   drawOval(RectF oval, Paint paint)
void   drawPaint(Paint paint)
void   drawPath(Path path, Paint paint)
void   drawPoint(float x, float y, Paint paint)
void   drawPoints(float[] pts, int offset, int length, Paint paint)
void   drawPoints(float[] pts, Paint paint)
void   drawPosText(char[] text, int index, int count, float[] pos,
Paint paint)
```

```
void    drawPosText(String text, float[] pos, Paint paint)
void    drawRGB(int r, int g, int b)
void    drawRect(RectF rect, Paint paint)
void    drawRect(float left, float top, float right, float bottom, Paint
paint)
void    drawRect(Rect r, Paint paint)
void    drawRoundRect(RectF rect, float rx, float ry, Paint paint)
void    drawText(String text, int start, int end, float x, float y, Paint
paint)
void    drawText(char[] text, int index, int count, float x, float y,
Paint paint)
void    drawText(String text, float x, float y, Paint paint)
void    drawText(CharSequence text, int start, int end, float x, float
y, Paint paint)
void    drawTextOnPath(String text, Path path, float hOffset, float
vOffset, Paint paint)
void    drawTextOnPath(char[] text, int index, int count, Path path,
float hOffset, float vOffset, Paint paint)
```

14.2 變形與轉換

Android 中二維的動畫功能主要在 `android.view.animation` 函式庫中，包含兩種動畫功能，第一種是二維轉換 (tweened animation)，第二種是逐畫面的動畫(frame by frame animation)。

二維轉換總共有四種基本轉換

1. `AlphaAnimation` 透明度轉換 (transparency changes)
2. `RotateAnimation` 旋轉 (rotations)
3. `ScaleAnimation` 縮放 (growing or shrinking)
4. `TranslateAnimation` 位移 (position changes)

描述二維轉換的方式又可分為 1. 使用 XML 2. 使用 Java 程式 兩種方法。

(1) 使用 XML 描述二維轉換

```
<set android:shareInterpolator="true"
      android:interpolator="@android:anim/ease_in_interpolator">
```

```

<translate android:fromXDelta="0"
          android:toXDelta="30"
          android:duration="800"
          android:fillAfter="true"/>

<set android:duration="800"
      android:pivotX="50%"
      android:pivotY="50%" >

    <rotate android:fromDegrees="0"
           android:toDegrees="-90"
           android:fillAfter="true"
           android:startOffset="800"/>

    <scale android:fromXScale="1.0"
           android:toXScale="2.0"
           android:fromYScale="1.0"
           android:toYScale="2.0"
           android:startOffset="800" />

```

</set>

```

<translate android:toYDelta="-100"
          android:fillAfter="true"
          android:duration="800"
          android:startOffset="1600"/>

```

</set>

(2) 使用 Java 實作二維轉換

```

// Create root AnimationSet.
AnimationSet rootSet = new AnimationSet(true);
rootSet.setInterpolator(new EaseInInterpolator());
rootSet.setRepeatMode(Animation.NO_REPEAT);

// Create and add first child, a motion animation.
TranslateAnimation trans1 = new TranslateAnimation(0, 30, 0, 0);
trans1.setStartOffset(0);
trans1.setDuration(800);
trans1.setFillAfter(true);

```

```
rootSet.addAnimation(trans1);

// Create a rotate and a size animation.
RotateAnimation rotate = new RotateAnimation(
    0,
    -90,
    RotateAnimation.RELATIVE_TO_SELF, 0.5f,
    RotateAnimation.RELATIVE_TO_SELF, 0.5f);
rotate.setFillAfter(true);
rotate.setDuration(800);

ScaleAnimation scale = new ScaleAnimation(
    1, 2, 1, 2, // From x, to x, from y, to y
    ScaleAnimation.RELATIVE_TO_SELF, 0.5f,
    ScaleAnimation.RELATIVE_TO_SELF, 0.5f);
scale.setDuration(800);
scale.setFillAfter(true);

// Add rotate and size animations to a new set,
// then add the set to the root set.
AnimationSet childSet = new AnimationSet(true);
childSet.setStartOffset(800);
childSet.addAnimation(rotate);
childSet.addAnimation(scale);
rootSet.addAnimation(childSet);

// Add a final motion animation to the root set.
TranslateAnimation trans2 = new TranslateAnimation(0, 0, 0, -100);
trans2.setFillAfter(true);
trans2.setDuration(800);
trans2.setStartOffset(1600);
rootSet.addAnimation(trans2);

// Start the animation.
animWindow.startAnimation(rootSet);
```

載入與啓動二維轉換

// Hook into the object to be animated.

```
TextView animWindow = (TextView)findViewById(R.id.anim);

// Load the animation from XML (XML file is
res/anim/move_animation.xml).
Animation anim =
AnimationUtils.loadAnimation(AnimationSample.this,
R.anim.move_animation);
anim.setRepeatMode(Animation.NO_REPEAT);

// Play the animation.
animWindow.startAnimation(anim);
```

14.3

第15章3D 動畫 OpenGL

Android 使用 OpenGL 這個 3D 動畫遊戲引擎的嵌入式版本，稱為 OpenGL ES，這和 J2ME 中的 JSR239 OpenGL ES API 大至是相同的，但並不完全一樣。

15.1 3D 動畫的基本概念

15.2 3D 模型的設計

15.3 OpenGL 的範例

在 Android 中使用 OpenGL 時，首先要先繼承 View 這個物件，然後在程式中取得 OpenGLContext 這個物件，接著在 `onDraw()` 這個函數中執行對應的繪圖的功能，以下是一個範例。

```
class GLView extends View
{
    public GLView(Context context)
    {
        ...
        mGLContext = new OpenGLContext(0);
        ...
    }
    @Override
    protected void onDraw(Canvas canvas) {
        ...
        GL10 gl = (GL10)(mGLContext.getGL());
        ...
        mGLContext.waitNative(canvas, this);
        ...
        gl.glViewport(0, 0, w, h);
        float ratio = (float)w / h;
        gl.glMatrixMode(gl.GL_PROJECTION);
        gl.glLoadIdentity();
```

```
gl.glFrustumf(-ratio, ratio, -1, 1, 2, 12);
gl.glDisable(gl.GL_DITHER);
gl.glClearColor(1,1,1,1);
gl.glEnable(gl.GL_SCISSOR_TEST);
gl.glScissor(0, 0, w, h);
gl.glClear(gl.GL_COLOR_BUFFER_BIT);
gl.glMatrixMode(gl.GL_MODELVIEW);
gl.glLoadIdentity();
gl.glTranslatef(0, 0, -3.0f);
gl.glScalef(0.5f, 0.5f, 0.5f);
gl.glRotatef(mAngle, 0, 1, 0);
gl.glRotatef(mAngle*0.25f, 1, 0, 0);
gl glColor4f(0.7f, 0.7f, 0.7f, 1.0f);
gl glEnableClientState(gl.GL_VERTEX_ARRAY);
gl glEnableClientState(gl.GL_COLOR_ARRAY);
gl glEnable(gl.GL_CULL_FACE);
mCube.draw(gl);
mAngle += 1.2f;
mGLContext.waitGL();

...
}

...
}
```

Android 的 `android.media.MediaPlayer` 函式庫提供了影片與聲音的播放功能，另外也在 `android.media.MediaRecorder` 中提供了錄音與錄影的功能，但這些功能在模擬器當中將無法運作，只有真實的手機才會具備此功能。

第16章影像與聲音功能

您可播放位於程式資源資料夾中的影音資源，也可以播放檔案系統中的影音檔，甚至播放網路上的影音檔案，但方法稍有差異。

16.1 播放內部影音資源

```
MediaPlayer mp = MediaPlayer.create(context, R.raw.sound_file_1);
mp.prepare();
mp.start();
```

16.2 播放外部影音檔案

```
MediaPlayer mp = new MediaPlayer();
mp.setDataSource(PATH_TO_FILE);
mp.prepare();
mp.start();
```

16.3 播放網路上的影音檔案

```
ContentURI myURL = new
ContentURI("http://myserver.com/link/to/my.mp3");
Intent intent = new Intent(Intent.VIEW_ACTION, myURL);
intent.setType("audio/*");
startActivity(intent);
```

16.4 摄取畫面

```
Bitmap copyWindowBitmap()
Return a copy of the bitmap holding the overall contents of the
```

window this view is attached to.

```
void getAbsoluteLocationOnScreen(int[] location)
Computes the coordinates of this view on the screen.
```

16.5 錄音與錄影

1. 啓動錄製功能

```
recorder = new MediaRecorder();
ContentValues values = new ContentValues(3);

values.put(Video.MediaColumns.TITLE, SOME_NAME_HERE);
values.put(Video.MediaColumns.TIMESTAMP,
System.currentTimeMillis());
values.put(Video.MediaColumns.MIME_TYPE,
recorder.getMimeType());

contentResolver = new ContentResolver();

ContentURI base = Video.Media.INTERNAL_CONTENT_URI;
ContentURI newUri = contentResolver.insert(base, values);

if (newUri == null) {
    // need to handle exception here - we were not able to create a
new
    // content entry
}

String path = contentResolver.getDataFilePath(newUri);

// could use setPreviewDisplay() to display a preview to suitable
View here

recorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_G
PP);
```

```
recorder.setVideoSize(176, 144); // QCIF
recorder.setVideoFrameRate(15);
recorder.setVideoEncoder(MediaRecorder.VideoEncoder.H263);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
recorder.setOutputFile(path);

recorder.prepare();
recorder.start();
```

2. 停止錄製功能

```
recorder.stop();
recorder.release();
```

16.6 照相功能

```
package test.camera;

import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.PixelFormat;
import android.hardware.CameraDevice;
import android.os.Bundle;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.Window;

// -----
-

public class CameraTest extends Activity
{
    @Override
    protected void onCreate(Bundle icicle)
    {
```

```
super.onCreate(icicle);

// Hide the window title.
requestWindowFeature(Window.FEATURE_NO_TITLE);

// Make sure to create a TRANSLUCENT window. This is
recquired
// for SurfaceView to work. Eventually this'll be done by
// the system automatically.
getWindow().setFormat(PixelFormat.TRANSLUCENT);

// Create our Preview view and set it as the content of our
// Activity
mPreview = new Preview(this);
setContentView(mPreview);
}

@Override
protected boolean isFullscreenOpaque() {
    // Our main window is set to translucent, but we know that we
will
    // fill it with opaque data. Tell the system that so it can
perform
    // some important optimizations.
    return true;
}

@Override
protected void onResume()
{
    // Because the CameraDevice object is not a shared resource,
    // it's very important to release it when the activity is
paused.
    super.onResume();
    mPreview.resume();
}
```

```
@Override
protected void onPause()
{
    // Start Preview again when we resume.
    super.onPause();
    mPreview.pause();
}

private Preview mPreview;
}

// -----
-

class Preview extends SurfaceView implements SurfaceHolder.Callback
{
    Preview(Context context) {
        super(context);

        // Install a SurfaceHolder.Callback so we get notified when
the
        // underlying surface is created and destroyed.
        mHolder = getHolder();
        mHolder.setCallback(this);
        mHasSurface = false;

        // In this example, we hardcode the size of the preview. In a
real
        // application this should be more dynamic. This guarantees
that
        // the underlying surface will never change size.
        mHolder.setFixedSize(320, 240);
    }
}
```

```
public void resume() {
    // We do the actual acquisition in a separate thread. Create
it now.

    if (mPreviewThread == null) {
        mPreviewThread = new PreviewThread();
        // If we already have a surface, just start the thread
now too.

        if (mHasSurface == true) {
            mPreviewThread.start();
        }
    }
}

public void pause() {
    // Stop Preview.

    if (mPreviewThread != null) {
        mPreviewThread.requestExitAndWait();
        mPreviewThread = null;
    }
}

public boolean surfaceCreated(SurfaceHolder holder) {
    // The Surface has been created, start our main acquisition
thread.

    mHasSurface = true;
    if (mPreviewThread != null) {
        mPreviewThread.start();
    }
    // Tell the system that we filled the surface in this call.
    // This is a lie to prevent the system to fill the surface for
us
    // automatically.
    // THIS IS REQUIRED because otherwise we'll access the
Surface object
    // from 2 different threads which is not allowed (And will
crash
```

```
// currently).
    return true;
}

public void surfaceDestroyed(SurfaceHolder holder) {
    // Surface will be destroyed when we return. Stop the
    preview.
    mHasSurface = false;
    pause();
}

public void surfaceChanged(SurfaceHolder holder, int format, int
w, int h) {
    // Surface size or format has changed. This should not happen
    in this
    // example.
}

// -----
-

class PreviewThread extends Thread
{
    PreviewThread() {
        super();
        mDone = false;
    }

    @Override
    public void run() {
        // We first open the CameraDevice and configure it.
        CameraDevice camera = CameraDevice.open();
        if (camera != null) {
            CameraDevice.CaptureParams param = new
            CameraDevice.CaptureParams();
        }
    }
}
```

```

        param.type = 1; // preview
        param.srcWidth      = 1280;
        param.srcHeight     = 960;
        param.leftPixel     = 0;
        param.topPixel      = 0;
        param.outputWidth   = 320;
        param.outputHeight  = 240;
        param.dataFormat    = 2; // RGB_565
        camera.setCaptureParams(param);
    }

    // This is our main acquisition thread's loop, we go
until
    // asked to quit.
    SurfaceHolder holder = mHolder;
    while (!mDone) {
        // Lock the surface, this returns a Canvas that can
        // be used to render into.
        Canvas canvas = holder.lockCanvas();

        // Capture directly into the Surface
        if (camera != null) {
            camera.capture(canvas);
        }

        // And finally unlock and post the surface.
        holder.unlockCanvasAndPost(canvas);
    }

    // Make sure to release the CameraDevice
    if (camera != null)
        camera.close();
}

public void requestExitAndWait() {
    // don't call this from PreviewThread thread or it a
}

```

```
guaranteed

    // deadlock!
    mDone = true;
    try {
        join();
    } catch (InterruptedException ex) { }

}

private boolean mDone;
}

SurfaceHolder mHolder;
private PreviewThread mPreviewThread;
private boolean mHasSurface;
}
```

第17章電話功能 Telephony

Android 的電話功能主要在 `android.telephony` 這個函式庫當中，提供撥打與接收電話的功能，其中最重要的兩個物件是 `IPhone` 和 `PhoneNumberUtils`。

17.1 電話相關函式庫

17.2 撥打電話

當你想要在程式中呼叫撥打電話的功能時，必需在 `AndroidManifest.xml` 檔案中將允許程式播打電話的權限打開，其指令如下

```
<uses-permission id="android.permission.CALL_PHONE" />
```

要建立 `IPhone` 物件，必需透過 `ServiceManager` 類別，以下是取得 `IPhone` 的方法：

```
private static IPhone getPhoneInterface() throws  
DeadObjectException {  
    IServiceManager sm = ServiceManagerNative.getDefault();  
    IPhone phoneService =  
    IPhone.Stub.asInterface(sm.getService("phone"));  
    return phoneService;  
}
```

一但取得 `IPhone` 物件後，即可使用 `call` 或 `dial` 來撥打電話，然後使用 `endCall` 來結束通話。

```
void call(String number) : Place a call to the number.  
void dial(String number) : Dial a number.  
void endCall(boolean hangupOnly) : End call or go to the Home  
screen
```

`dial` 與 `call` 之間的差別是 `dial` 會顯示一個撥號介面，其上填入 `number` 這個電話號碼（若 `number` 是 `null`，則撥號介面中將不會有預設的號碼）。

另一種撥打電話的方法是對一個電話網址 (例如：tel:0988077312) 送出 CALL_ACTION 的動作，即可進行通話供能。

另外、你也可以使用 DataStateIntentReceiver 或 PhoneStateIntentReceiver 進行電話事件的註冊，以便取得撥號狀態 (IDLE, RINING, OFF_HOOK)、服務狀態 (in service, out of service, emergency only, powered off, roaming, operator name ...)、訊號強度、語音留言、連線狀態 (disconnected、connecting、connected)、資料進出狀態 (data in, data out) 等等。

17.3 取得電話資訊

您也可以透過 TelephonyProperties 中的屬性值，取得許多手機相關的資訊，例如手機的電話號碼、SIM 卡的資訊等等，其方法是使用 os.SystemProperties.get() 函數，傳入 TelephonyProperties 中的對應參數即可取得之，反之 若使用 os.SystemProperties.put() 則可設定這些參數。

範例一：取得 IMEI 國際手機代碼

```
android.os.SystemProperties.get(PROPERTY imei)
```

範例二：取得手機的電話號碼

```
android.os.SystemProperties.get(PROPERTY_line1_number)
```

範例三：取得手機的語音郵件號碼

```
android.os.SystemProperties.get(PROPERTY_line1_voice_mail_number)
```

範例四：取得電信公司的名稱

```
android.os.SystemProperties.get(PROPERTY_sim_operator_alpha)
```

範例五：取得國家的代碼

```
android.os.SystemProperties.get(PROPERTY_sim_operator_iso_country)
```

17.4 簡訊功能

17.5 E-mail 功能

第18章衛星定位

與衛星定位相關得兩個函式庫是 android.location 與 com.google.android.maps，目前還再草案狀態未定案

18.1 衛星 GPS 相關函式庫

18.2 衛星定位的程式

android.location

預計將支援四種地圖相關資訊 (class、kml、nmea、track)，其中的 LocationManager 與 LocationProvider 是核心元件。

要取得 Location Manager 的方式必需透過 Context，方法如下：

LocationManager lm =

Context.getSystemService(Context.LOCATION_SERVICE);

```
try {
    Intent myIntent = new
Intent(android.content.Intent.VIEW_ACTION,
    new ContentURI("geo:38.899533,-77.036476"));
    startActivityForResult(myIntent);
} catch (URISyntaxException e) { }
```

18.3 控制 Google Map 的顯示

com.google.android.maps

此函式庫支援 GoogleMap 應用程式的控制，您可以透過 MapView 物件顯示 GoogleMap，但前題是您的 Activity 必需繼承 MapActivity。

接著您可以透過 MapView 中的 getController() 或 getOverlayController()

去控制MapView。

```
package test.map1;

import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import android.os.Bundle;

public class MapViewTest1 extends MapActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        MapView mapView = new MapView(this);
        MapController mc = mapView.getController();
        mc.zoomTo(9);
        setContentView(mapView);
    }
}
```



```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <view class="com.google.android.maps.MapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1" />

    <EditText android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="InputAddress"
        android:selectAllOnFocus="true"/>
</LinearLayout>
```

```
package test.map;

import com.google.android.maps.MapActivity;
import android.os.Bundle;

public class MapViewTest extends MapActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
    }
}
```



18.4 在 GoogleMap 上重疊顯示

<http://davanum.wordpress.com/2007/11/19/drawing-overlays-for-android-maps-aka-search-for-starbucks/>

```
package org.apache.maps;

import android.graphics.Canvas;
import android.graphics.Paint;
import com.google.android.maps.Overlay;
import com.google.android.maps.Point;
import com.google.googlenav.Placemark;
import com.google.googlenav.Search;

public class MyOverlay extends Overlay {

    BrowseMap mMap;
    Paint paint1 = new Paint();
    Paint paint2 = new Paint();

    public MyOverlay(BrowseMap map) {
        mMap = map;
        paint2.setARGB(255, 255, 255, 255);
    }

    public void draw(Canvas canvas, PixelCalculator
pixelCalculator, boolean b) {
        super.draw(canvas, pixelCalculator, b);

        Search search = mMap.getSearch();
        if (search != null) {
            for (int i = 0; i < search.numPlacemarks(); i++) {
                Placemark placemark = search.getPlacemark(i);
                int[] screenCoords = new int[2];
                Point point = new
Point(placemark.getLocation().getLatitude(),
                    placemark.getLocation().getLongitude());
                pixelCalculator.getPointXY(point, screenCoords);
                canvas.drawCircle(screenCoords[0], screenCoords[1],
9, paint1);
                canvas.drawText(Integer.toString(i + 1),
screenCoords[0] - 4,
```

```
    screenCoords[1] + 4, paint2);  
}  
}  
}  
}
```

第19章藍芽功能

藍芽裝置由於具有獨特的搜尋裝置等特性，因此、某些功能無法納入普通的 TCP/IP 網路功能之下，因此、需要特殊的函式庫支援，在 Android 中，採用的是開放原始碼的 bluez 函式庫 (在 Android 中路徑為 org.bluez)。

19.1 開放原始碼的藍芽函式庫 - BlueZ

19.2 搜尋藍芽裝置

19.3 檔案與物件交換

19.4 透過藍芽連上 Internet

第20章整合範例

20.1

20.2 檔案瀏覽器

第21章自製視覺化元件

<http://code.google.com/android/toolbox/custom-components.html>

您可以經由繼承 view, layout, 或任何從 View 衍生下來的元件以自製，

These steps provide a high level overview of what you need to know to get started in creating your own components:

1. Extend an existing View class or subclass with your own class.
2. Override some of the methods from the superclass: the superclass methods to override start with 'on', for example, onDraw(), onMeasure(), and onKeyDown().
This is similar to the on... events in Activity or ListActivity that you override for life cycle and other functionality hooks.
3. Use your new extension class: once completed, your new extension class can be used in place of the view upon which it was based, but now with the new functionality

The CustomView sample in the API Demos provides an example of a customized component. The custom component is defined in the LabelView class.

21.1

21.2

第22章部署與安裝程式

22.1 建立安裝程式

22.2 部署程式到 Google Phone 上

第23章 Google Phone 的未來

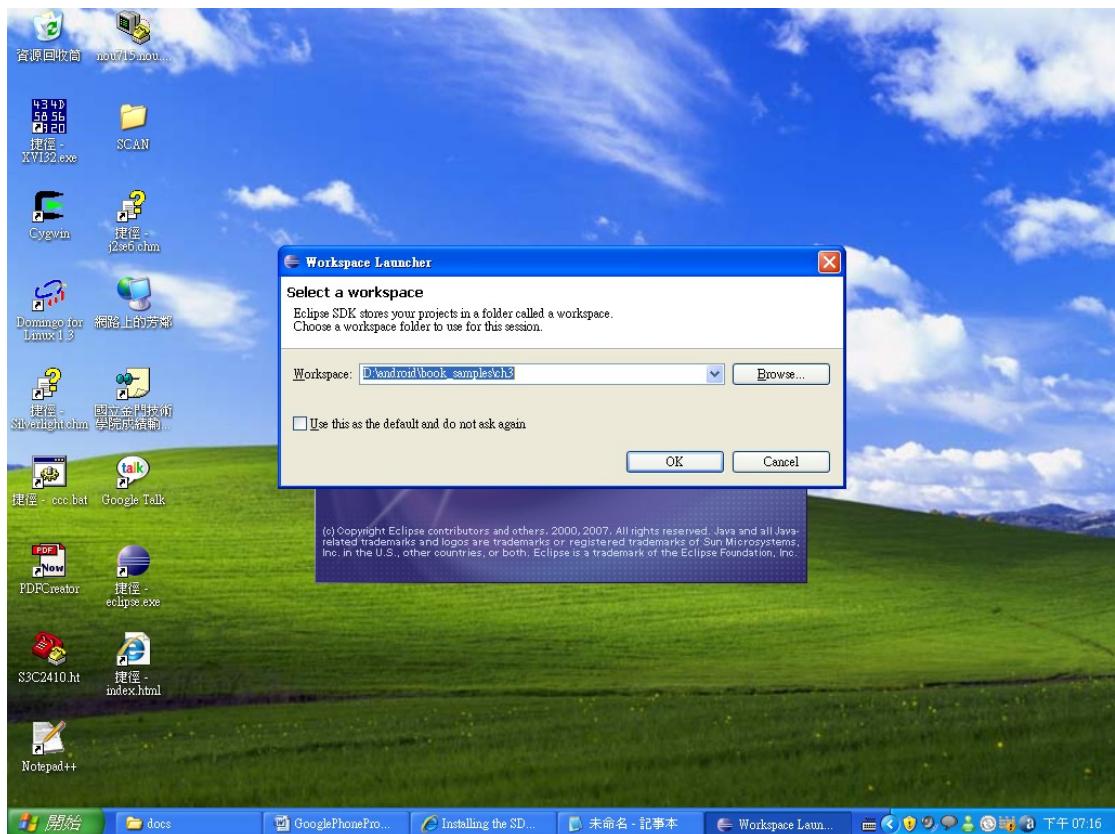
23.1 Google

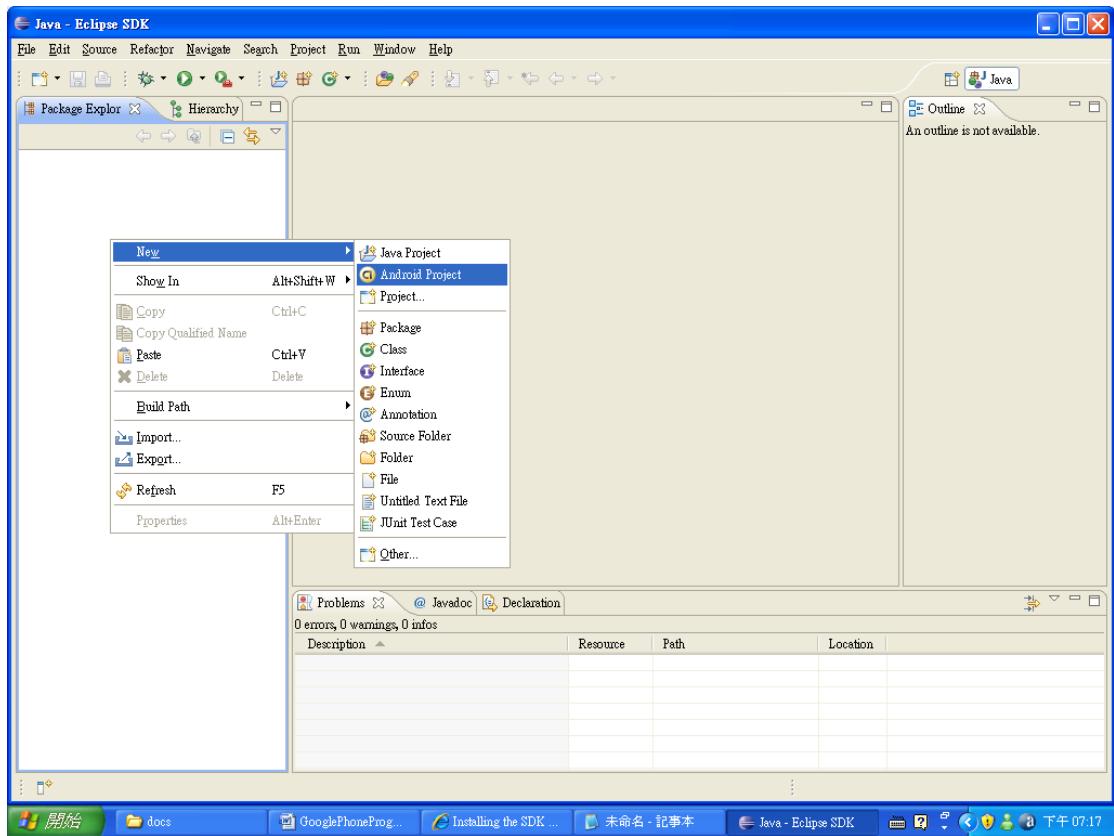
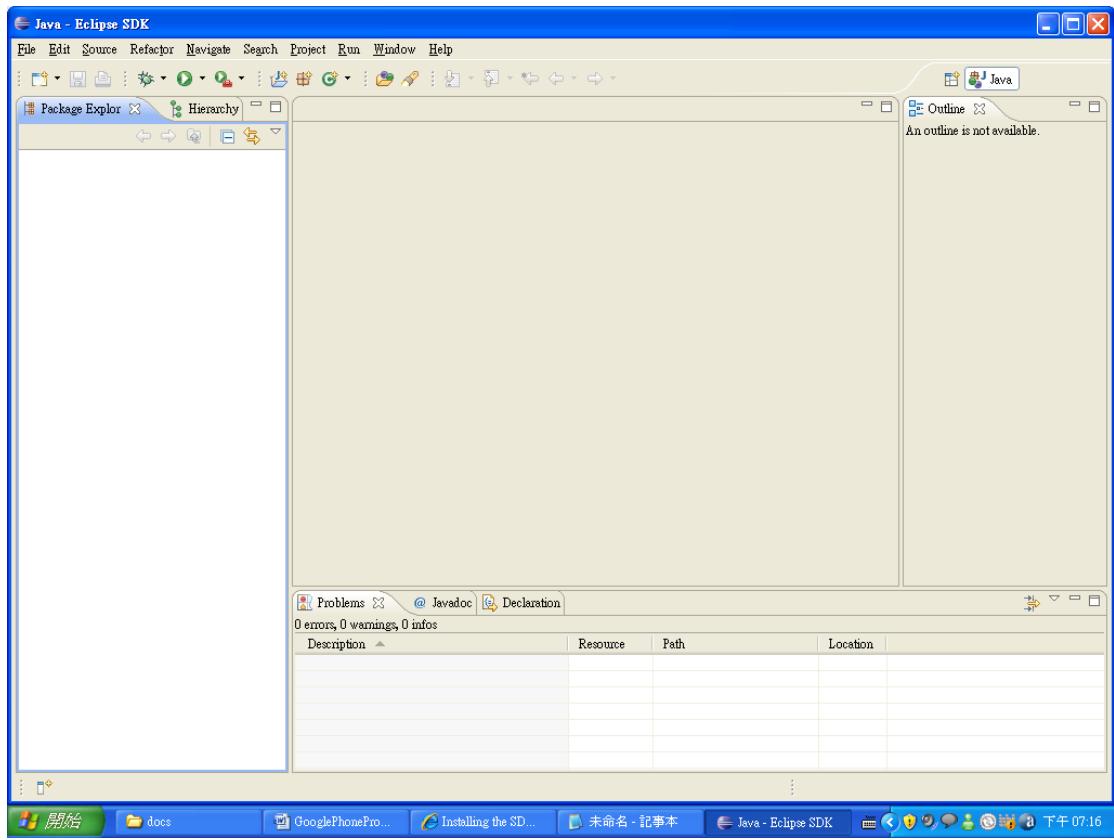
23.2 結論

附錄 1Android 的開發工具

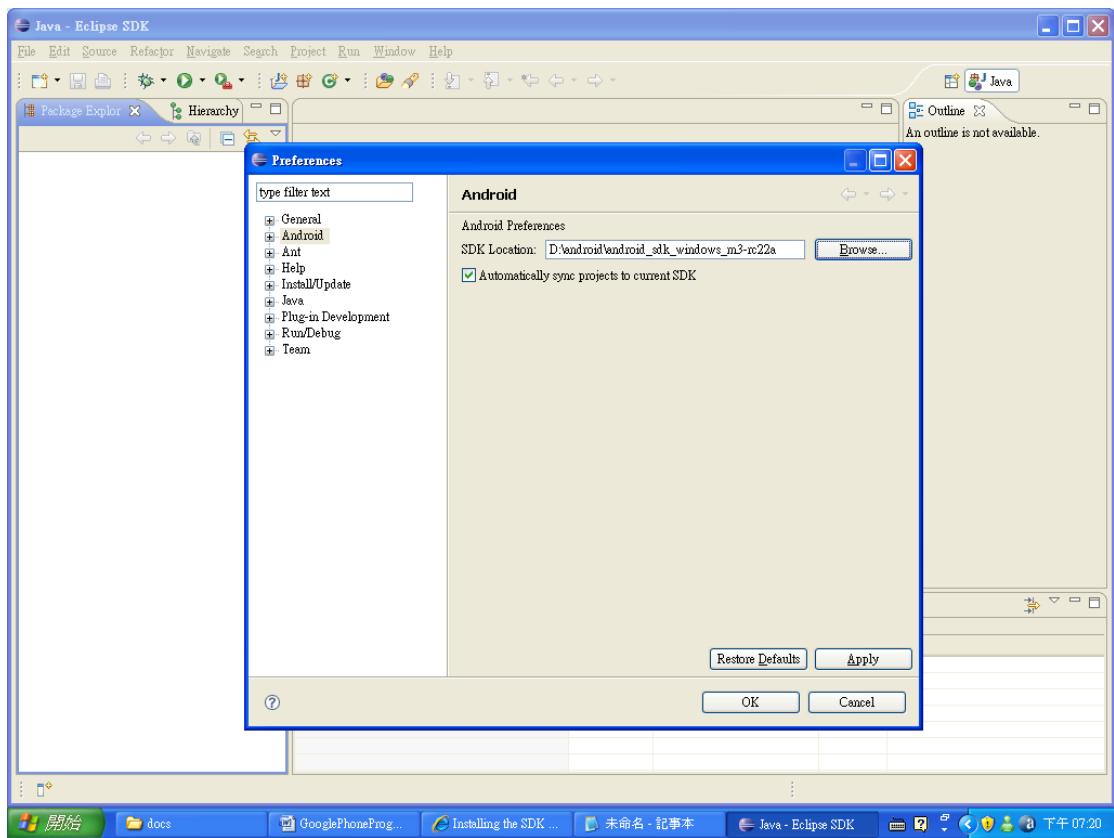
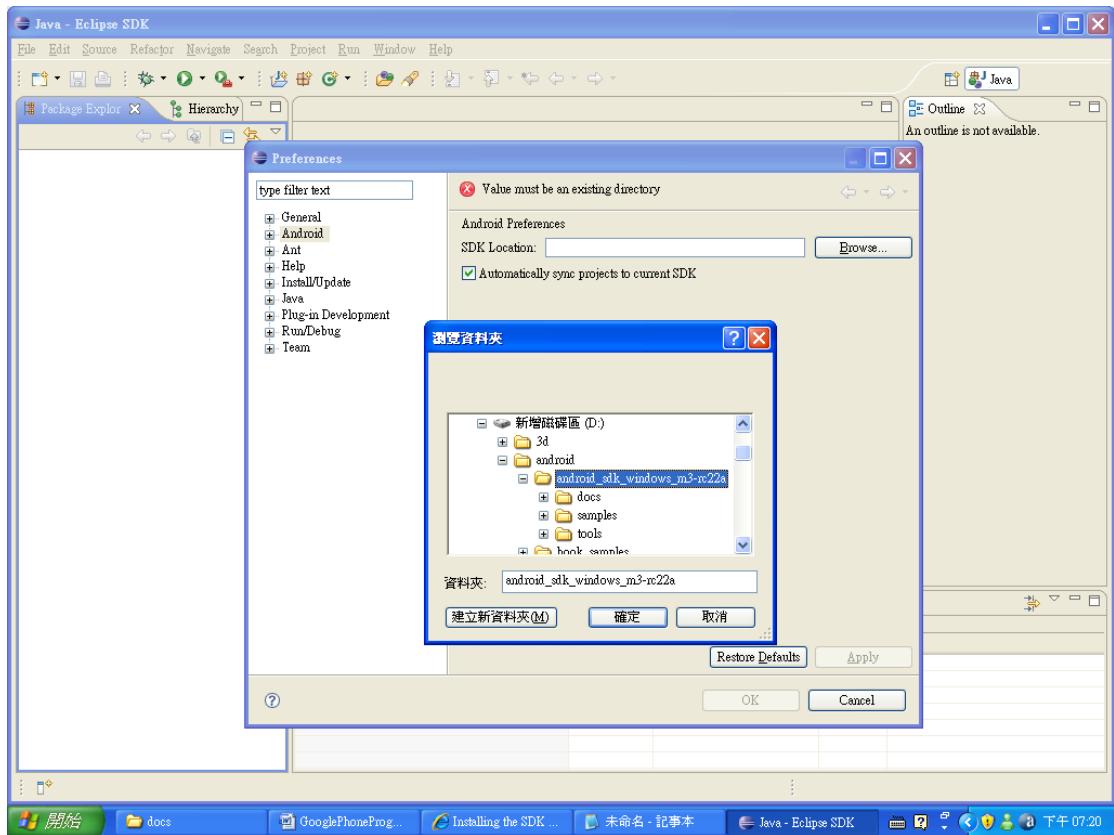
附錄2Eclipse 的使用方法

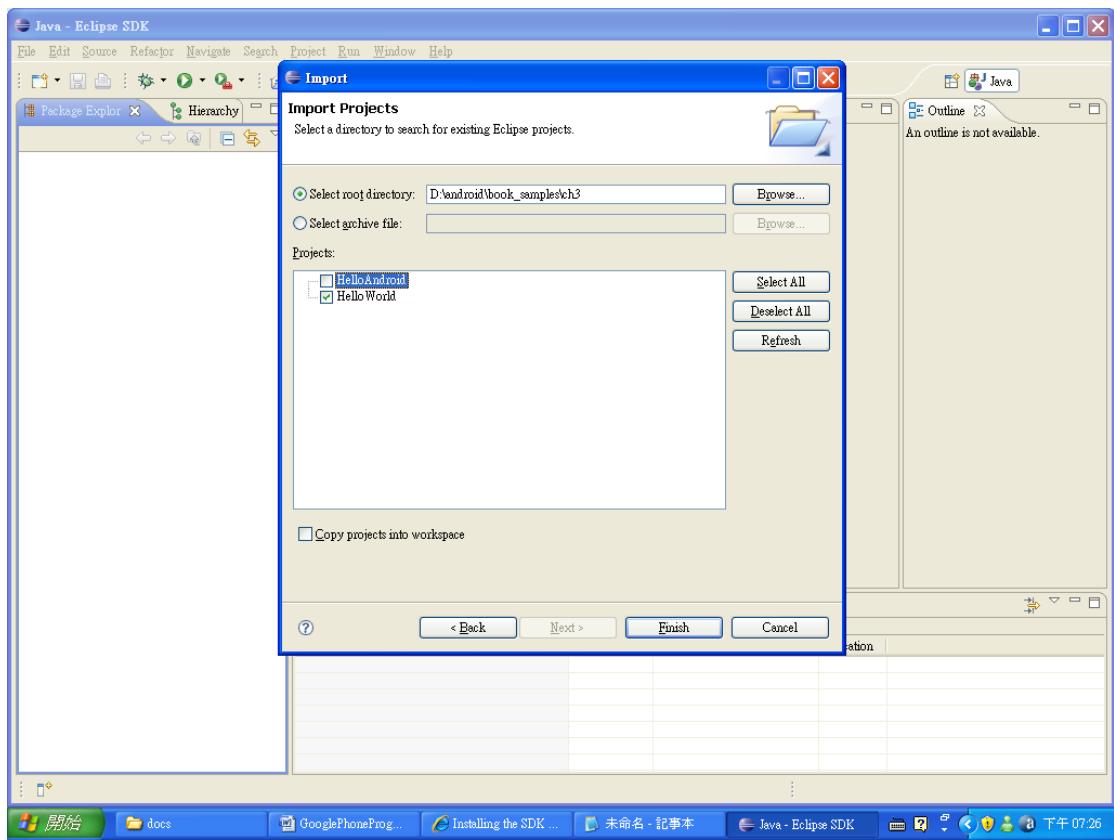
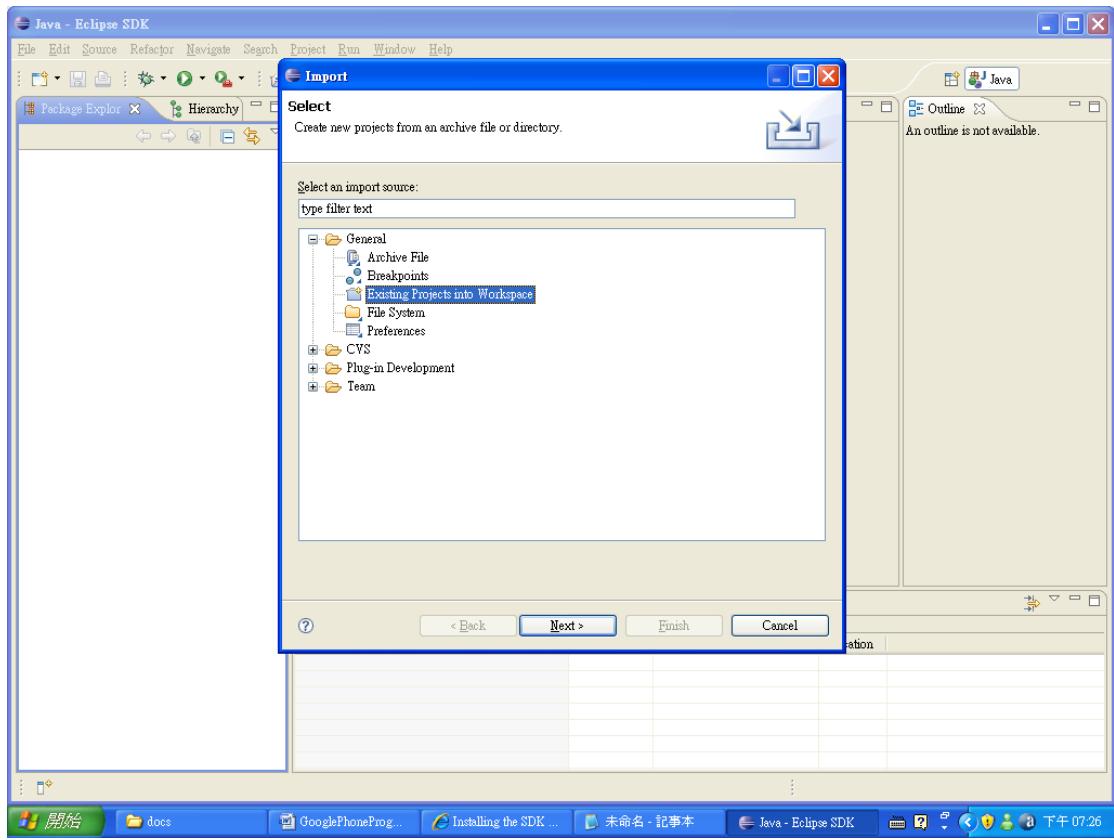
附錄3本書範例程式的使用方法





Windows/preferences





附錄4Adb 除錯工具的使用

```
D:\ccc\code\Android\sdk\tools>adb
-d <device number>           - directs command to a specific device
devices                      - list all connected devices

device commands:
adb update DATAOPTS <file>  - Flash the specified update file.
                                If file is not passed, update.zip is used.
adb push <local> <remote>   - copy file/dir to device
adb pull <remote> <local>    - copy file/dir from device
adb sync [ <localdir> ]      - copy host->device only if changed
                                (see 'adb help all')
adb shell                   - run remote shell interactively
adb shell <command>         - run remote shell command
adb logcat [ <filter-spec> ] - View device log
adb forward <local> <remote> - forward socket connections
                                forward specs are one of:
                                tcp:<port>
                                local:<unix domain socket name>
                                dev:<character device name>
adb install <app>          - push this app to the data partition
adb bugreport                - return all information from the device
                                that should be included in a bug report.

adb help                     - show this help message

DATAOPTS:
(no option)                 - don't touch the data partition
-w                         - wipe the data partition
-d                         - flash the data partition

bootloader commands:
adb flashall DATAOPTS       - reflash the device from the build
```

output tree

```
adb flash [<name>] [<file>] - write to flash  
adb send <name> <file>      - write to ram  
adb debug                  - listen to bootloader debuglog  
adb bl <command>          - send raw bootloader command
```

scripting:

```
adb wait-for-bootloader    - block until bootloader is online  
adb wait-for-device       - block until device is online  
adb start-server          - ensure that there is a server running  
adb kill-server            - kill the server if it is running  
adb get-state              - prints: offline | bootloader | device  
adb get-product            - prints: <product-id>  
adb get-serialno           - prints: <serial-number>
```

networking:

adb ppp <tty> [parameters] - Run PPP over USB.

Note: you should not automatically start a PDP connection.

<tty> refers to the tty for PPP stream. Eg.

dev:/dev/omap_csmi_tty1

[parameters] - Eg. defaultroute debug dump local notty

usepeerdns

adb sync notes: adb sync [<localdir>]

<localdir> can be interpreted in several ways:

- If it not specified, both /system and /data partitions will be updated.

- If it is "system" or "data", only the corresponding partition is updated.

- If it is a path to a local directory, the name is examined to see if it points to a directory named ".../system" or ".../data", or if the directory contains subdirectories with those names. If so, it pushes

the appropriate directory/ies to the device.

- If the name of the local directory does not match ".../system" or ".../data", it is treated like an "system" directory.
- If <localdir> points to a nonexistent directory, adb sync will fail.

```
D:\ccc\code\Android\sdk\tools>adb shell
# sqlite3
/data/data/com.google.android.providers.settings/databases/settings.db

sqlite3
/data/data/com.google.android.providers.settings/databases/settings.db
SQLite version 3.5.0
Enter ".help" for instructions
sqlite> delete from system where _id=99
delete from system where _id=99
...> ;
;
sqlite> ls
ls
...> ;
;
SQL error: near "ls": syntax error
sqlite> .help
.help
.bail ON|OFF      Stop after hitting an error. Default OFF
.databases        List names and files of attached databases
.dump ?TABLE? ... Dump the database in an SQL text format
.echo ON|OFF       Turn command echo on or off
.exit             Exit this program
.explain ON|OFF    Turn output mode suitable for EXPLAIN on or off.
.header(s) ON|OFF  Turn display of headers on or off
```

```
.help          Show this message
.import FILE TABLE Import data from FILE into TABLE
.indices TABLE    Show names of all indices on TABLE
.load FILE ?ENTRY? Load an extension library
.mode MODE ?TABLE? Set output mode where MODE is one of:
                  csv   Comma-separated values
                  column Left-aligned columns. (See .width)
                  html   HTML <table> code
                  insert SQL insert statements for TABLE
                  line   One value per line
                  list   Values delimited by .separator string
                  tabs   Tab-separated values
                  tcl    TCL list elements
.nullvalue STRING Print STRING in place of NULL values
.output FILENAME   Send output to FILENAME
.output stdout     Send output to the screen
.prompt MAIN CONTINUE Replace the standard prompts
.quit              Exit this program
.read FILENAME     Execute SQL in FILENAME
.schema ?TABLE?    Show the CREATE statements
.separator STRING   Change separator used by output mode and
.import
.show              Show the current values for various settings
.tables ?PATTERN?  List names of tables matching a LIKE pattern
.timeout MS        Try opening locked tables for MS milliseconds
.width NUM NUM ... Set column widths for "column" mode
sqlite> select * from system
select * from system
...> ;
;
1|music_volume|3
2|voice_volume|3
3|ringer_volume|2
4|ringer_volume_last_positive_index|2
sqlite> .exit
.exit
```

```
# .exit
.exit
.exit: not found
# .quit
.quit
.quit: not found
# .help
.help
.help: not found
# ^C
D:\ccc\code\Android\sdk\tools>
```

附錄5設定 Android 中 Proxy 的方法

在程式中設定

```
public static void setProxy() {  
  
    Properties prop = System.getProperties();  
    prop.setProperty("http.proxyHost", "proxy.internal");  
    prop.setProperty("http.proxyPort", "3128");  
  
    prop.setProperty("https.proxyHost", "proxy.internal");  
    prop.setProperty("https.proxyPort", "3128");  
  
    prop.setProperty("ftp.proxyHost", "proxy.internal");  
    prop.setProperty("ftp.proxyPort", "3128");  
    prop.setProperty("ftp.nonProxyHosts", "localhost|10.10.*");  
  
    prop.setProperty("socksProxyHost", "proxy.internal");  
    prop.setProperty("socksProxyPort", "3128");  
}
```

在系統中設定

Here is a potential work around. It uses the command line to add an [http proxy](#).

Note: you may need to update the escaping to use for windows.

cd to the location of adb

replace the pieces [host_or_IP] and [port] with the correct values for the proxy. So if your proxy is corp_proxy, on port 8080, then in the quotes should be corp_proxy:8080

[./adb shell sqlite3 /data/data/com.google.android.providers.settings/](#)

```
databases/settings.db "\""INSERT INTO system  
VALUES(99,'http_proxy','[host_or_IP]:[port]');\""
```

To see if it was added correctly you can run the following command line. You should see 99|http_proxyl [host_or_IP]:[port]

```
./adb shell sqlite3 /data/data/com.google.android.providers.settings/  
databases/settings.db "\""SELECT * FROM system\""
```

To remove the proxy, the following script can be run:

```
./adb shell sqlite3 /data/data/com.google.android.providers.settings/  
databases/settings.db "\""DELETE FROM system WHERE _id=99\""
```

附錄6Android 的函式庫列表

android	Contains the resource classes used by standard Android applications.
android.app	High-level classes encapsulating the overall Android application model.
android.content	Contains classes for accessing and publishing data on the device.
android.database	Contains classes to explore data returned through a content provider.
android.database.sqlite	Contains the SQLite database management classes that an application would use to manage its own private database.
android.graphics	Provides low level graphics tools such as canvases, color filters, points, and rectangles that let you handle drawing to the screen directly.
android.graphics.dra	Provides classes to manage a variety of

wable	visual elements that are intended for display only, such as bitmaps and gradients.
android.graphics.glutils	Provides a variety of classes to enable using OpenGL for embedded systems (OpenGL ES) to draw graphics on an Android device.
android.hardware	Provides support for hardware devices that may not be present on every Android device. android.location Classes defining Android location-based and related services.
android.media	
android.net	Classes that help with network access, beyond the normal java.net.* APIs.
android.opengl	Provides OpenGL utilities.
android.os	Provides basic operating system services, message passing, and inter-process communication on the device.
android.provider	Provides convenience classes to access the content providers supplied by Android.
android.sax	A framework that makes it easy to write efficient and robust SAX handlers.
android.speech.recognition	Provides classes for speech recognition.
android.telephony	Provides tools to make, receive, and monitor phone calls and phone status.
android.telephony.gsm	Provides classes to control or read data from GSM phones.
android.text	Provides classes used to render or track text and text spans on the screen.
android.text.method	Provides classes that monitor or modify keypad input.
android.text.style	Provides classes used to view or change the style of a span of text in a View object.
android.text.util	
android.util	Provides common utility methods such as date/time manipulation, base64 encoders and decoders, string and number conversion methods, and XML utilities.
android.view	Provides classes that expose basic user interface classes that handle screen layout and interaction with the user.
android.view.animation	Provides classes that handle tweened animations.
android.webkit	Provides tools for browsing the web.
android.widget	The widget package contains (mostly visual) UI elements to use on your Application screen.

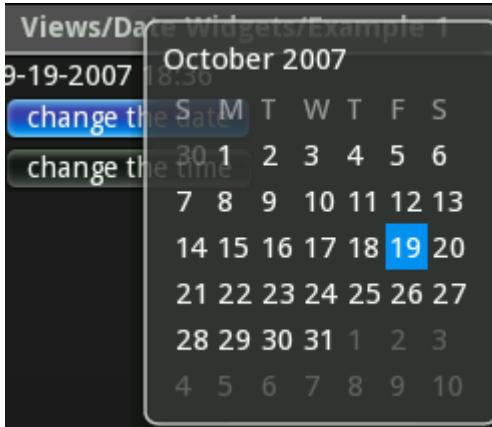
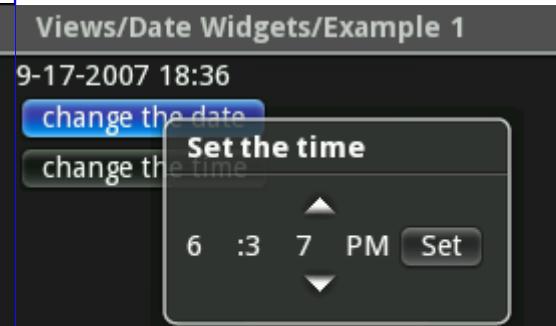
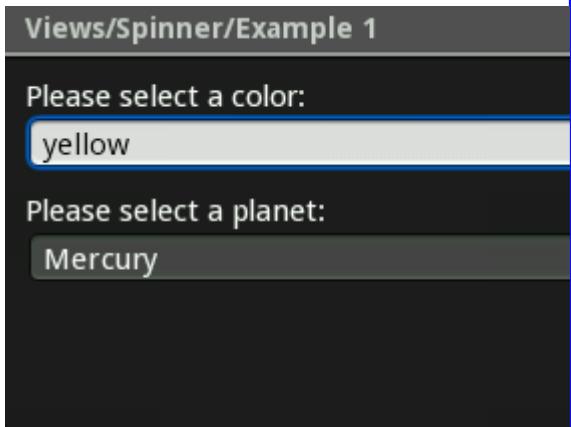
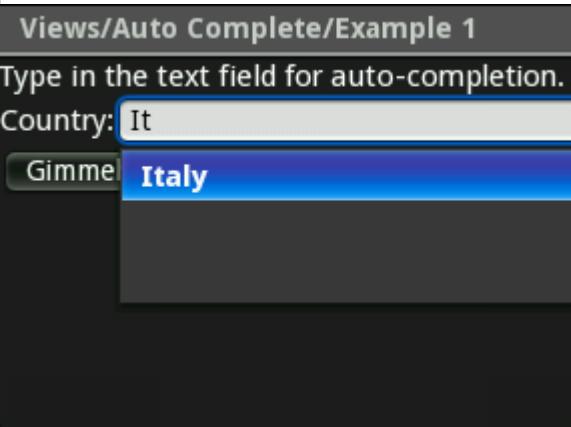
com.google.android.maps	
com.google.android.xmppService	
java.io	
java.lang	
java.lang.annotation	
java.lang.instrument	
java.lang.ref	
java.lang.reflect	
java.math	
java.net	
java.nio	
java.nio.channels	
java.nio.channels.spi	
java.nio.charset	
java.nio.charset.spi	
java.security	
java.security.acl	
java.security.cert	
java.security.interfaces	
java.security.spec	
java.sql	
java.text	
java.util	
java.util.concurrent	Utility classes commonly useful in concurrent programming.
java.util.concurrent.atomic	A small toolkit of classes that support lock-free thread-safe programming on single variables.
java.util.concurrent.locks	Interfaces and classes providing a framework for locking and waiting for conditions that is distinct from built-in synchronization and monitors.
java.util.jar	
java.util.logging	
java.util.prefs	
java.util.regex	
java.util.zip	
javax.crypto	
javax.crypto.interfaces	

es	
javax.crypto.spec	
javax.microedition.k ronos.opengles	
javax.net	
javax.net.ssl	
javax.security.auth	
javax.security.auth.c allback	
javax.security.auth.lo gin	
javax.security.auth.x 500	
javax.security.cert	
javax.sound.midi	
javax.sound.midi.spi	
javax.sound.sampled	
javax.sound.sampled. .spi	
javax.sql	
javax.xml.parsers	
junit.extensions	
junit.framework	
org.apache.common s.codec	A small set of interfaces used by the various implementations in the sub-packages.
org.apache.common s.codec.binary	Base64, Binary, and Hexadecimal String encoding and decoding.
org.apache.common s.codec.language	Language and phonetic encoders.
org.apache.common s.codec.net	Network related encoding and decoding.
org.apache.common s.httpclient	Classes and interfaces supporting the client side of the HTTP protocol.
org.apache.common s.httpclient.auth	Provides implementation of various authentication schemes as well as utility classes that can be used to authenticate HTTP requests.
org.apache.common s.httpclient.cookie	Provides cookie handling in conjunction with Cookie.
org.apache.common s.httpclient.methods	Classes implementing HttpMethod for the base HTTP methods.
org.apache.common s.httpclient.methods. multipart	Multipart support classes for the MultipartPostMethod. Provides

org.apache.common s.httpclient.params	HttpClient preferences framework.
org.apache.common s.httpclient.protocol	Provides protocol specific socket factory handling.
org.apache.common s.httpclient.util	Provides some utility classes for use by HttpClient.
org.bluez	Provides classes to manage Bluetooth functionality on the device.
org.json	
org.w3c.dom	
org.xml.sax	This package provides the core SAX APIs.
org.xml.sax.ext	This package contains interfaces to SAX2 facilities that conformant SAX drivers won't necessarily support.
org.xml.sax.helpers	This package contains "helper" classes, including support for bootstrapping SAX-based applications.

附錄7Android 中的視覺畫元件展示館

附錄8下列畫面顯示了 Android 中主要的視覺化元件的呈現外觀，這些範例在 Android SDK 中的 AdiDemo 專案中都有展示。

<p>DatePicker</p> <p>Sample: DateWidgets.java</p> 	<p>TimePicker</p> <p>Sample: DateWidgets.java</p> 
<p>Spinner</p> <p>Sample: Spinner1.java</p> 	<p>AutoComplete</p> <p>Sample: AutoComplete1.java</p> 

[Button](#) [EditText](#) [CheckBox](#)

[Radio](#)

Sample: [Controls2.java](#)

Views/Controls/Example 2

[Save](#)

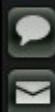
- Checkbox 1
- Checkbox 2
- RadioButton 1
- RadioButton 2

Ash

[ImageButton](#)

Sample: [ImageButton1.java](#)

Views/ImageButton/Example 1



[Gallery](#)

Sample: [Gallery1.java](#)

[ImageSwitcher](#)

Sample: [ImageSwitcher1.java](#)

Views/Gallery/1. Photos



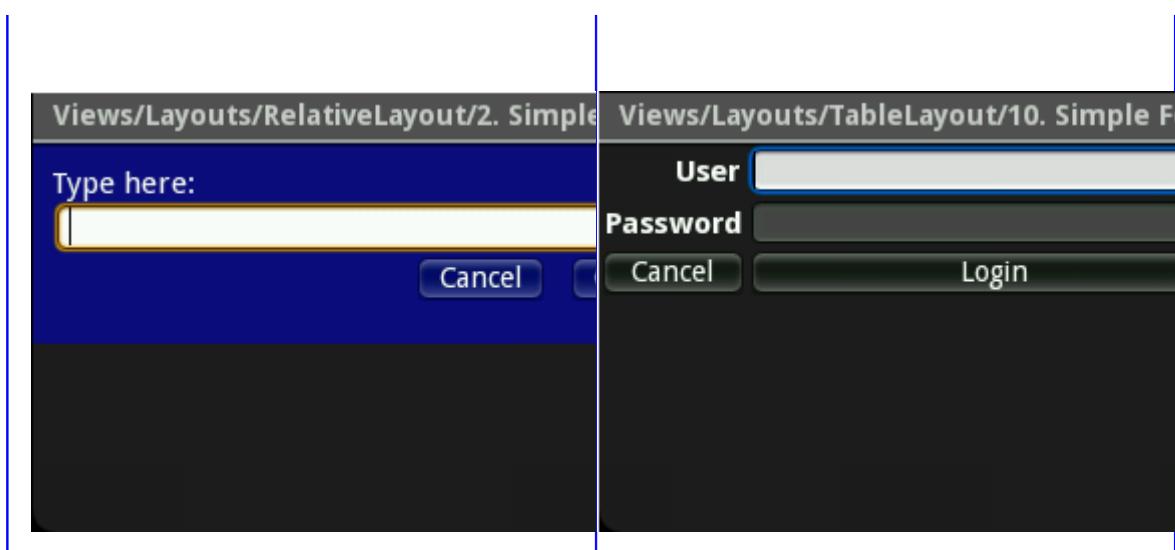
[ListView](#)

Sample: [List1.java](#)

[GridView](#)

Sample: [Grid2.java](#)

<p>Views/Lists/Example 1</p> <p>Abbaye de Belloc</p> <p>Abbaye du Mont des Cats</p> <p>Abertam</p> <p>Abondance</p> <p>Ackawi</p> <p>Acorn</p> <p>Adelost</p> <p>Affidolice au Chablis</p>	<p>Views/Grid/2. Photo Grid</p> 				
<p>LinearLayout (horizontal)</p> <p>Sample: LinearLayout1.java</p>	<p>LinearLayout (vertical)</p> <p>Sample: LinearLayout7.java</p>				
<p>Views/Layouts/LinearLayout/Example 7</p> <table border="1" data-bbox="230 1066 785 1448"> <tr> <td>Not much text</td> <td>A lot more text than any of the other columns. This column should set the height of the linear</td> <td>Not much text</td> <td>wrap_content</td> </tr> </table>	Not much text	A lot more text than any of the other columns. This column should set the height of the linear	Not much text	wrap_content	<p>Views/Layouts/LinearLayout/1. Vertical</p> <p>This is the top view.</p> <p>This is the middle view. It has more text in it than either the top view or the bottom view.</p> <p>This is the bottom view.</p>
Not much text	A lot more text than any of the other columns. This column should set the height of the linear	Not much text	wrap_content		
<p>RelativeLayout</p> <p>Sample: RelativeLayout2.java</p>	<p>TableLayout</p> <p>Sample: TableLayout10.java</p>				



附錄9Android 說明文件的閱讀指引

每個 Android 文件都包含下列組成，其中的 **XML Attributes** 就是記載了其對映 XML 屬性的描述，在摘要與詳細內容兩段都有，整個文件的結構如下：

1. 類別定義 ex : class android.widget.TableRow.LayoutParams
2. See Also : 記載相關類別的資訊
3. Summary : 摘要
 - XML Attributes
 - XML Attributes inherited from class ...
 - Constants inherited from class ...
 - Fields
 - Fields inherited from class ...
 - Public Constructors
 - ...
 - Protected Methods
 - Methods inherited from class
4. Detail : 詳細內容
 - XML Attributes
 - XML Attributes inherited from class ...
 - Constants inherited from class ...
 - Fields
 - Fields inherited from class ...
 - Public Constructors
 - ...
 - Protected Methods
 - Methods inherited from class

[android.widget](#)

public static class

android.widget.TableRow.LayoutParams

[java.lang.Object](#)

[android.view.ViewGroup.LayoutParams](#)

[android.view.ViewGroup.MarginLayoutParams](#)

This set of layout

[android.widget.LinearLayout.LayoutParams](#)

[android.widget.TableRow.LayoutParams](#)

parameters enforces the width of each child to be [FILL_PARENT](#) and the height of each child to be [WRAP_CONTENT](#).

See Also

- [TableLayout.LayoutParams](#)

Summary

XML Attributes

Attribute name

Related methods

[android:layout_column](#)

The index of the column in which this child should be.

[android:layout_span](#)

Defines how many columns this child should span.

⊕ XML Attributes inherited from class

[android.widget.LinearLayout.LayoutParams](#)

[android:layout_gravity](#), [android:layout_weight](#)

Attribute name

Related methods

[android:layout_gravity](#)

Standard gravity
constant that a child
can supply to its
parent.

[android:layout_weight](#)

⊕XML Attributes inherited from class

[android.view.ViewGroup.MarginLayoutParams](#)

[android:layout_marginBottom](#), [android:layout_marginLeft](#),
[android:layout_marginRight](#), [android:layout_marginTop](#)

Attribute name**Related methods**

[android:layout_marginBottom](#)

[setMargins\(int,int,int,int\)](#)

Specifies extra space on the bottom side of this view.

[android:layout_marginLeft](#)

[setMargins\(int,int,int,int\)](#)

Specifies extra space on the left side of this view.

[android:layout_marginRight](#)

[setMargins\(int,int,int,int\)](#)

Specifies extra space on the right side of this view.

[android:layout_marginTop](#)

[setMargins\(int,int,int,int\)](#)

Specifies extra space on the top side of this view.

[⊕](#)**XML Attributes inherited from class**

[**android.view.ViewGroup.LayoutParams**](#)

[android:layout_height](#), [android:layout_width](#)

Attribute name

Related methods

[android:layout_height](#)

Specifies the basic height of the view.

[android:layout_width](#)

Specifies the basic width of the view.

[⊕](#)**Constants inherited from class**

[**android.view.ViewGroup.LayoutParams**](#)

[FILL_PARENT](#), [WRAP_CONTENT](#)

v

alu

e

int

E

ILL

P

AR

EN

I

Sp

eci

al

val

ue

for

the

hei

ght

or

wid

th

req

ue

ste

d

by

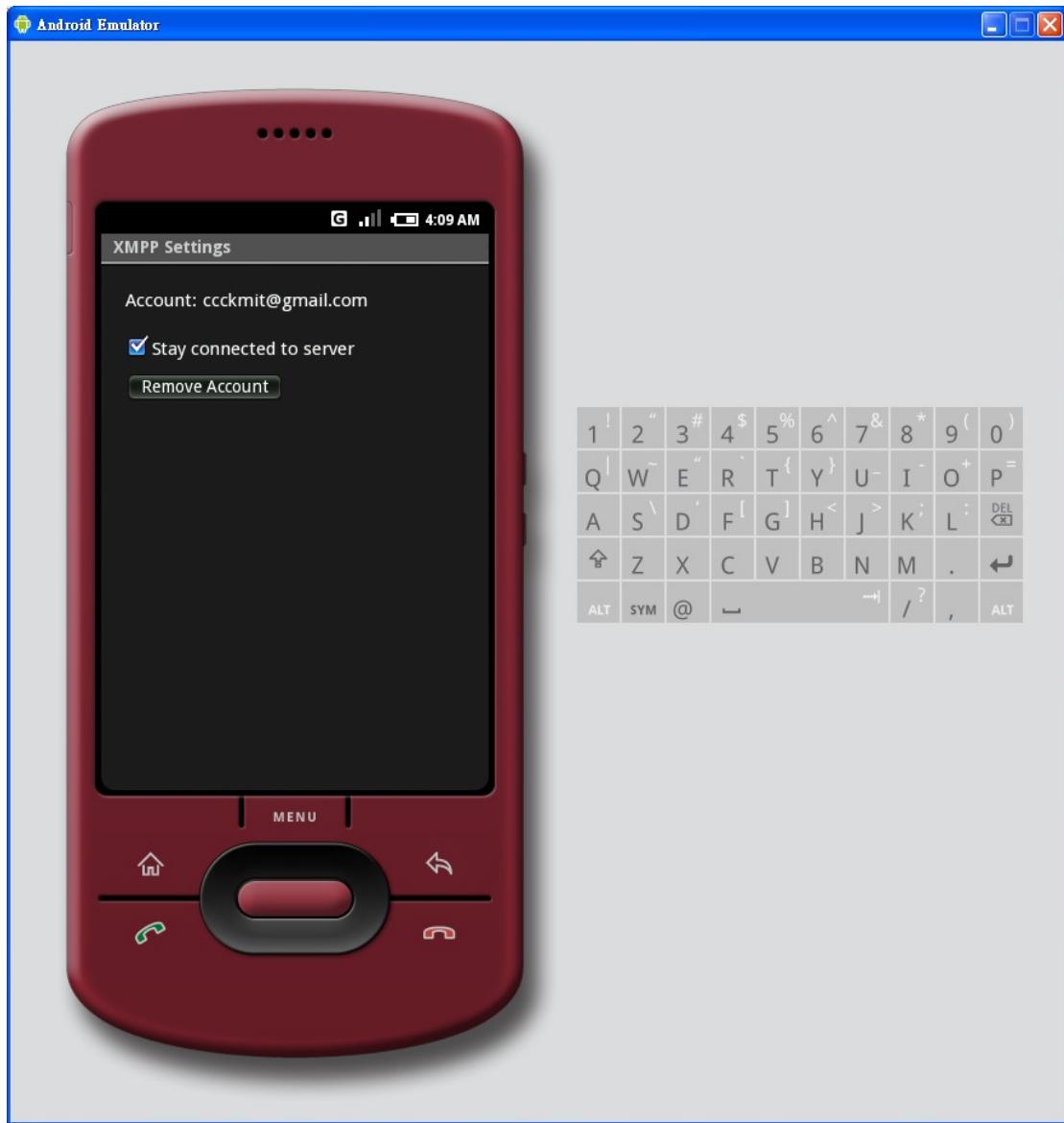
a

附錄10Android 與 Java 之間的關係

Android 的 DVM 所採用的是 register based machine，與 Sun 的 Stack Based machine 不同，而 Android 本身並不自稱 Java，但實際上卻是 100 % 的 Java，Android 為了迴避 Sun 採用了一些迂迴策略，

附錄11常見的錯誤狀況與解決方式

要連接 Google Map 請設定 XMPP Setting，加入你的 Gmail Account



Hey guys

I've been peacefully coding for about a week now with only a few problems, but right now out of goddamn nowhere the emulator started giving me that exception out of nowhere:

Java:

```
An error has occurred in process
com.google.process.content. Unable to
start receiver
com.google.android.xmppService.ServiceAutoStart:
java.lang.NullPointerException.
```

I changed only little in my code so I don't believe it is really my fault 😊

Now I cannot even run the Browser and the Maps without getting an NullPointerException
😢

Help pls..

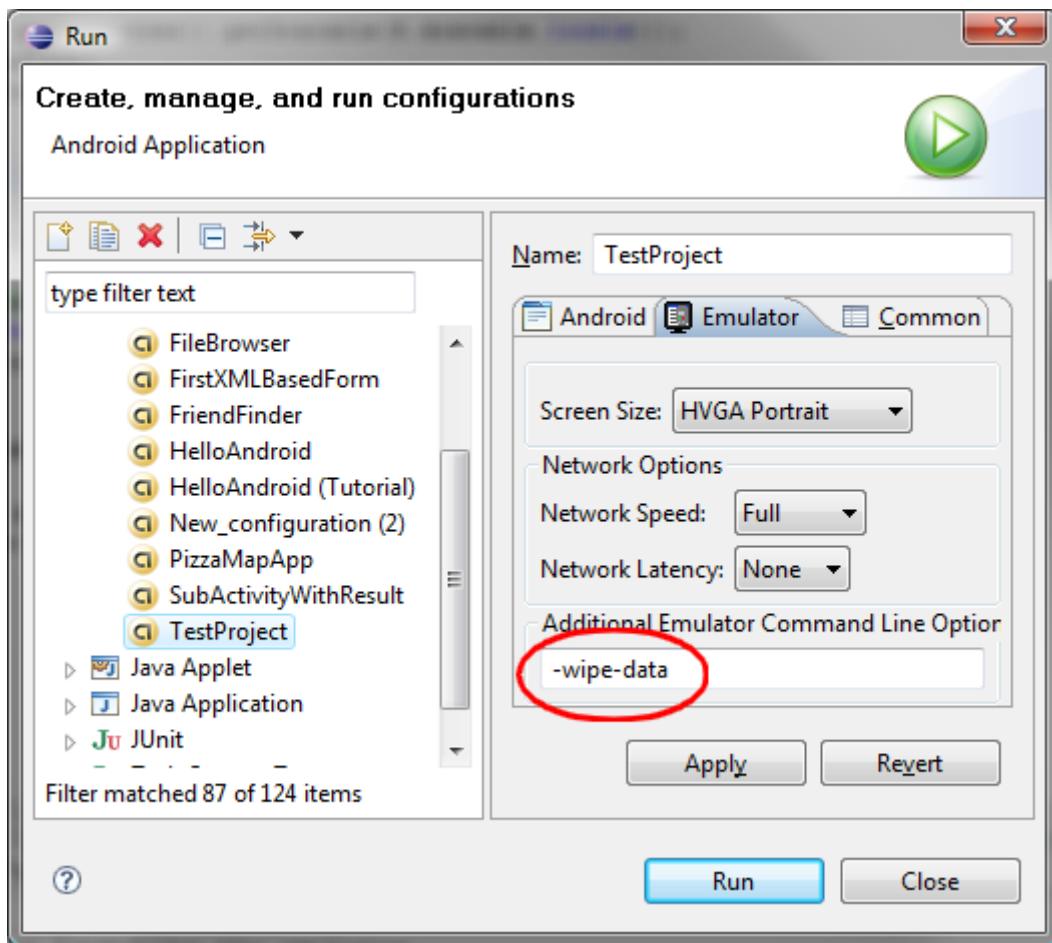
mrocket

Hello mrocket,

Up to now there are only two possible solutions for this problem.

*On both you should **save your data** (database / other files - BEFORE !)*

1.) Start the emulator once with "-wipe-data" on Additional Cmd-Line-Options:



or 2.) Delete the emulator-image (`userdata.img`) located at:

WinXP: "C:\Documents and Settings\<win_loginname>\Local Settings\Application Data\Android\"

WinVista: "C:\Users\<win_loginname>\AppData\Local\Android\"

Both should fix it!

Regards,
plusminus



| Android Development Community / Tutorials

附錄12Android 的 核程式碼

Download : <http://code.google.com/p/android/downloads/list>

How to Make :

http://blog.chinaunix.net/u/30686/showart_433547.html

download zip file(android_sdk_linux_m3-rc22a.zip) and then install

my linux machine(FC7)

I could see kernel's image in ..tools/lib/images directory

file name : kernel-qemu, size : 1.2M

1. <http://code.google.com/p/android/downloads/list>

download linux-2.6.23-android-m3-rc20.tar.gz android linux kernel
source

and uncompress kernel source

2. goldfish architecture configuration file copy to .config

goldfish configuration file in kernel/arch/arm/configs/
goldfish_defconfig

3. toolchain

http://www.codesourcery.com/gnu_toolchains/arm/download.html

checking radio box

Target Platform : ARM uClinux

Host Platform : IA32 GNU/LINUX

and then download...

4. uncompress arm-2007q3-51-arm-uclinuxeabi-i686-pc-linux-gnu.tar

after download

5. environment value setting /etc/bashrc

PATH=\$PATH:<arm-uclinuxeabi-* path>

export PATH

6. kernel compiled by arm-uclinuxeabi-* compiler

"make zImage"

7. kernel image(zImage) copy to <android>/tools/lib/images/

8. execute "emulator -console -debug-kernel -kernel zImage"

9. rename kernel image(zImage) to kernel-qemu

and execute "emulator -console -debug-kernel -kernel kernel-qemu"

注意：工具鏈裡面的 gcc 一定要選用 GCC 3.3.* ,GCC 4.2.* 有問題。

參考文獻

1. Android - An Open Handset Alliance Project - Open Source Licensing
<http://code.google.com/android/kb/licensingandoss.html>

2. Ryan Paul, Why Google chose the Apache Software License over GPLv2 for Android, Published: November 06, 2007 - 09:26AM CT
<http://arstechnica.com/news.ars/post/20071106-why-google-chose-the-apache-software-license-over-gplv2.html>

3. Installing the SDK

<http://code.google.com/android/intro/installing.html>

4. Download the Android SDK

<http://code.google.com/android/download.html>

5. Package Index

<http://code.google.com/android/reference/packages.html>

6. Android 中文網 <http://www.androidcn.net/>

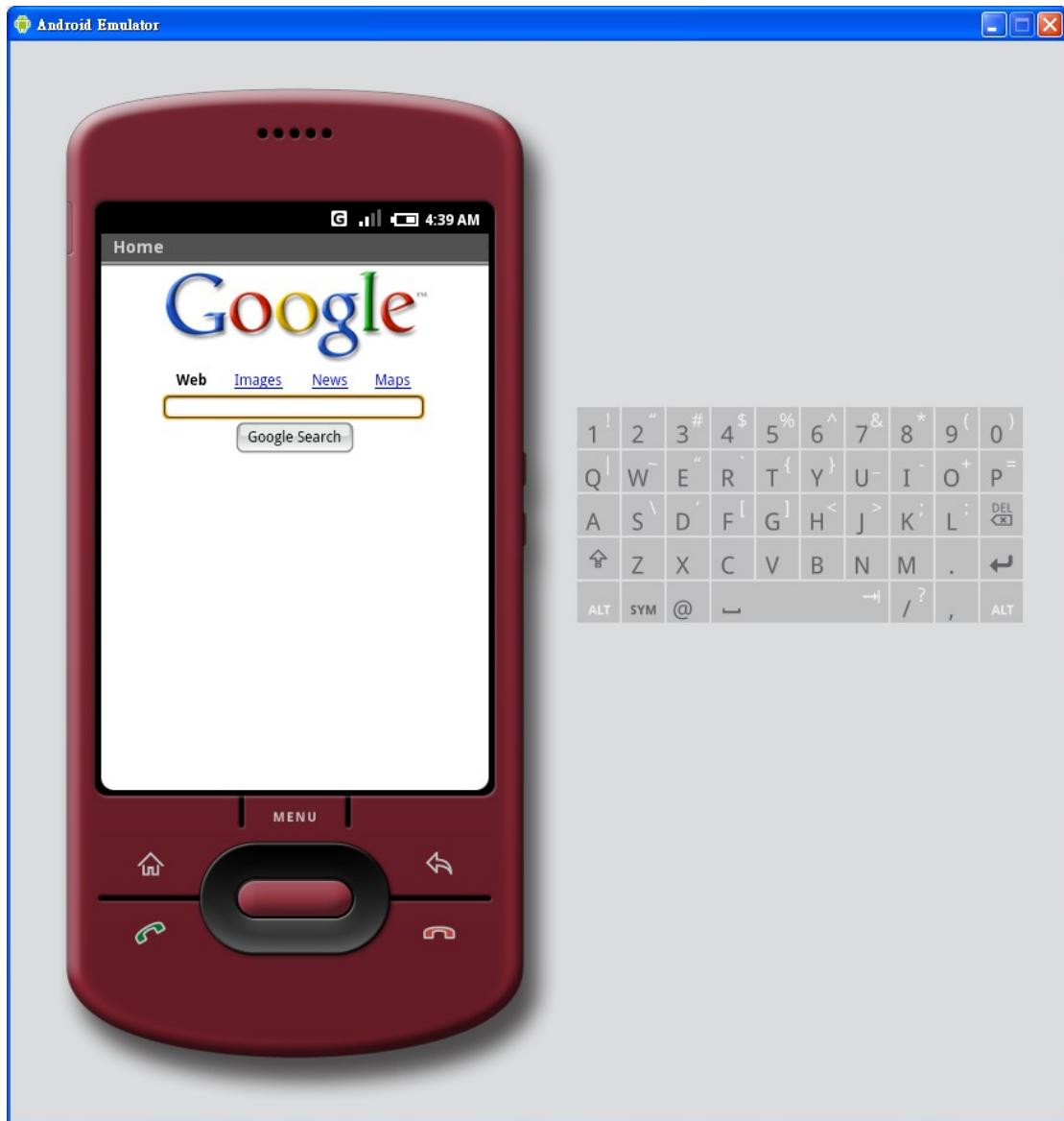
7. Android 中文網- Documentation

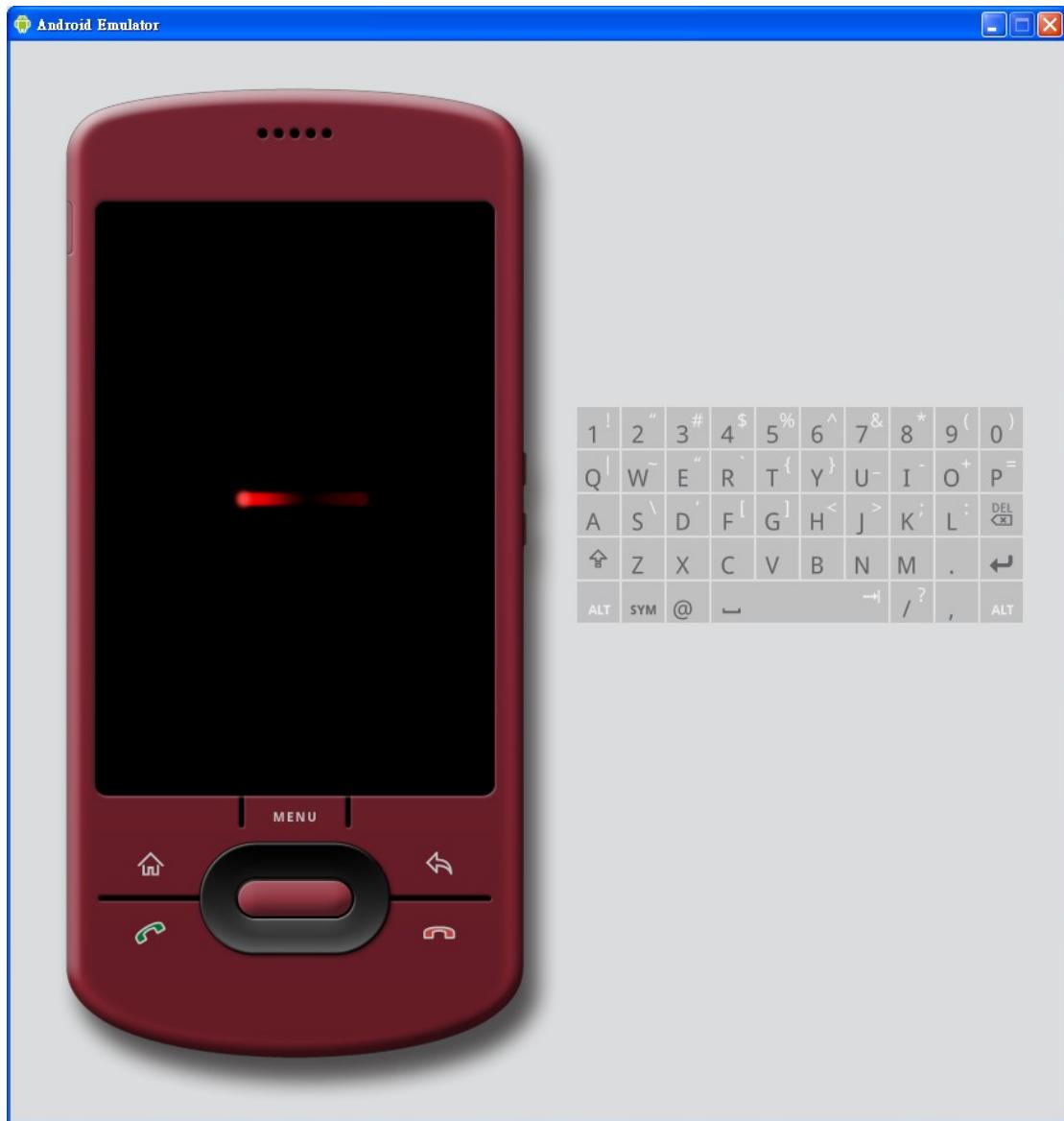
<http://www.androidcn.net/wiki/index.php/Documentation>

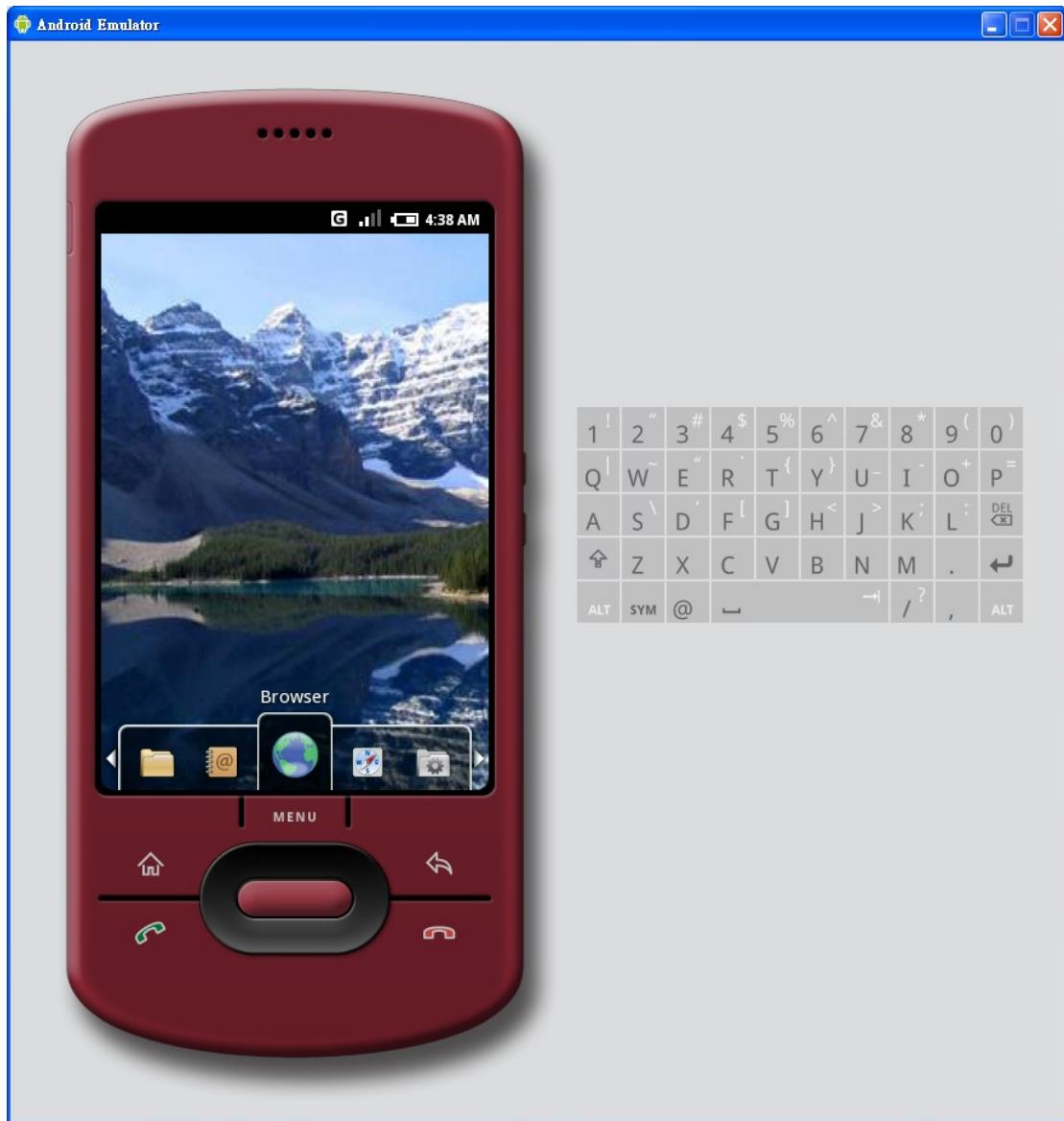


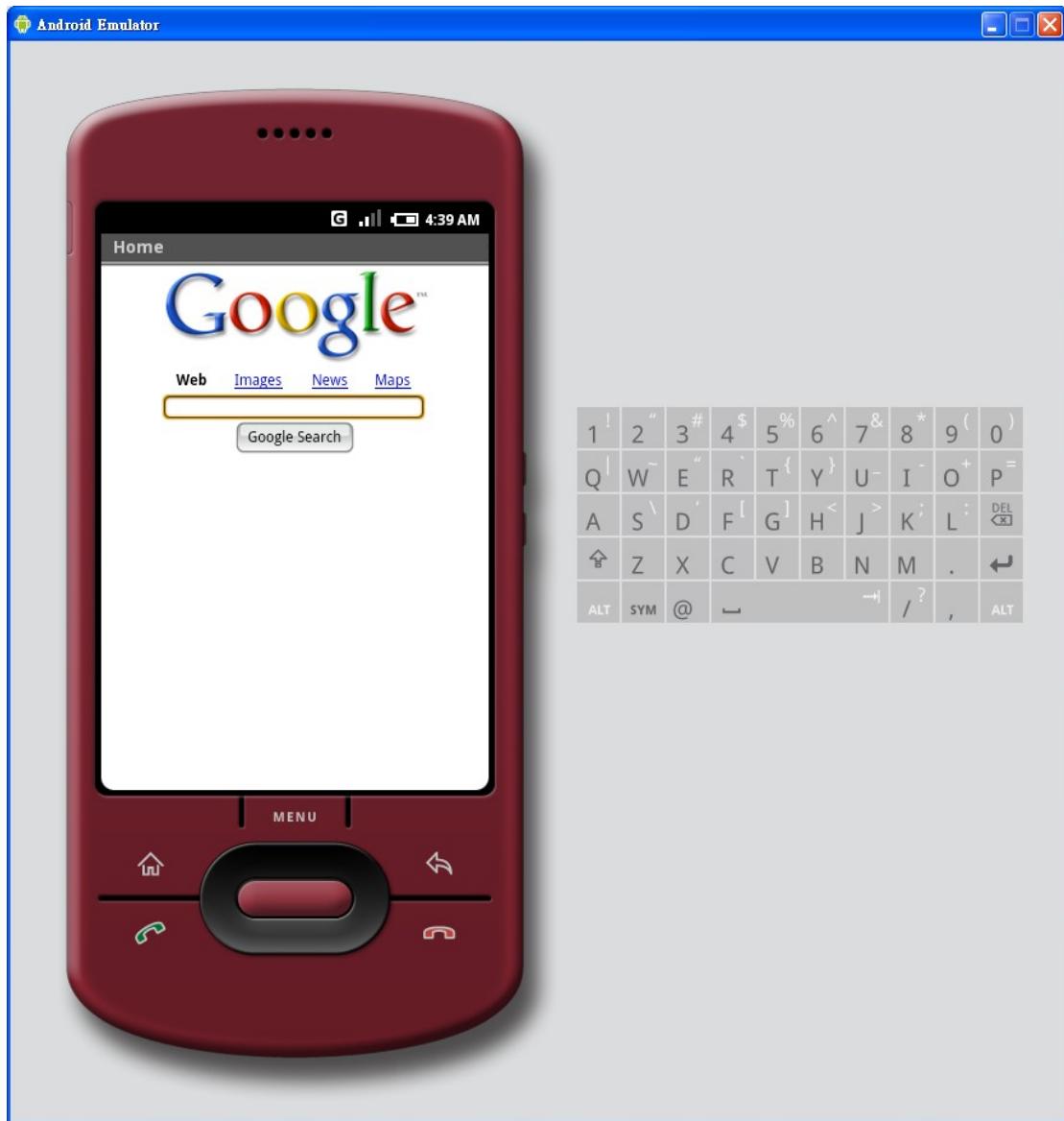
Google手機 程式設計

1	!	2	"	3	#	4	\$	5	%	6	^	7	&	8	*	9	()	0
Q	W	~	E	"	R	T	{	Y	}	U	-	I	-	O	+	P	=		
A	S	\	D	'	F	[G	H	<	J	>	K	:	L	:	DEL			
空格	Z	X	C	V	B	N	M	.									←		

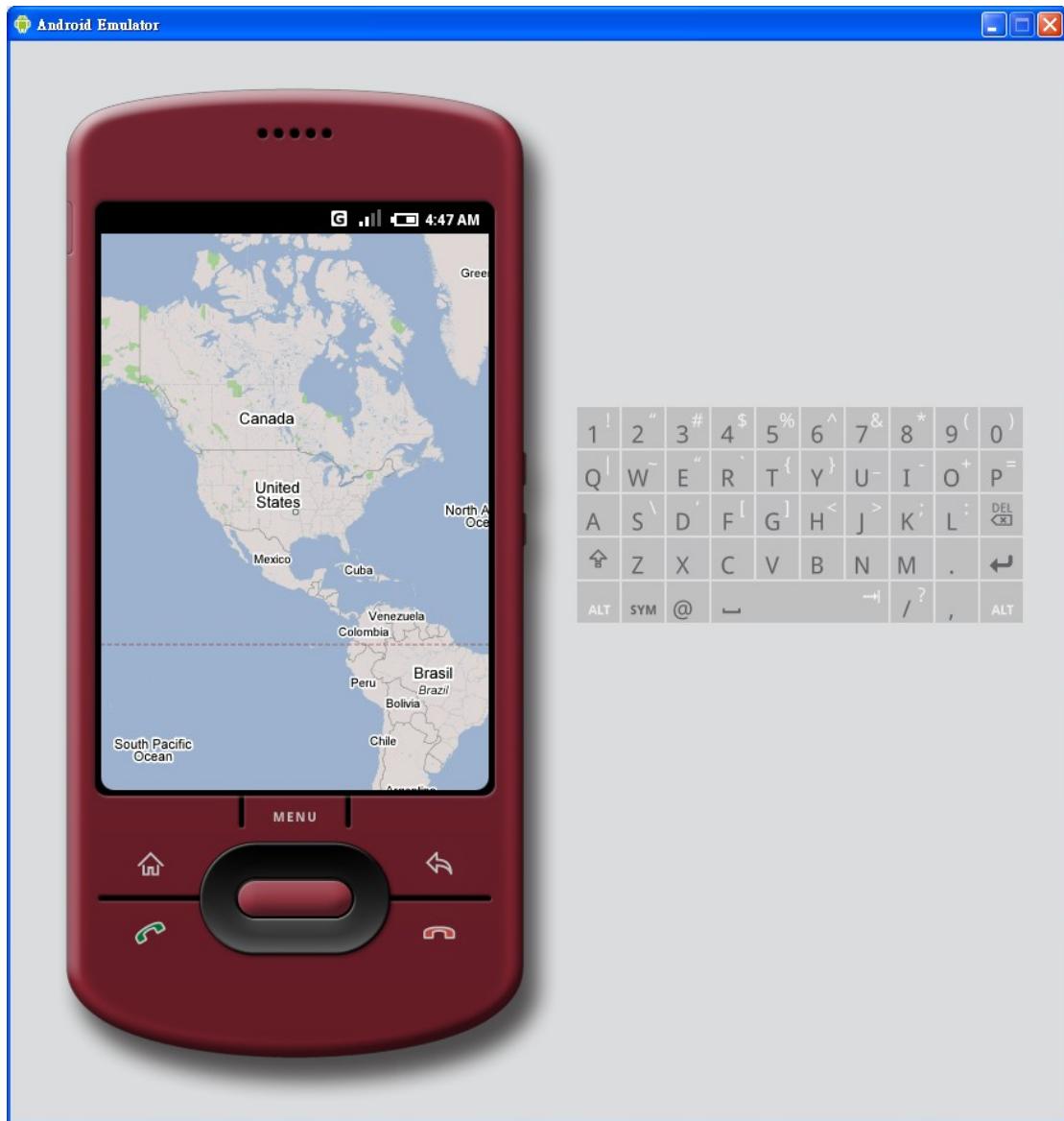


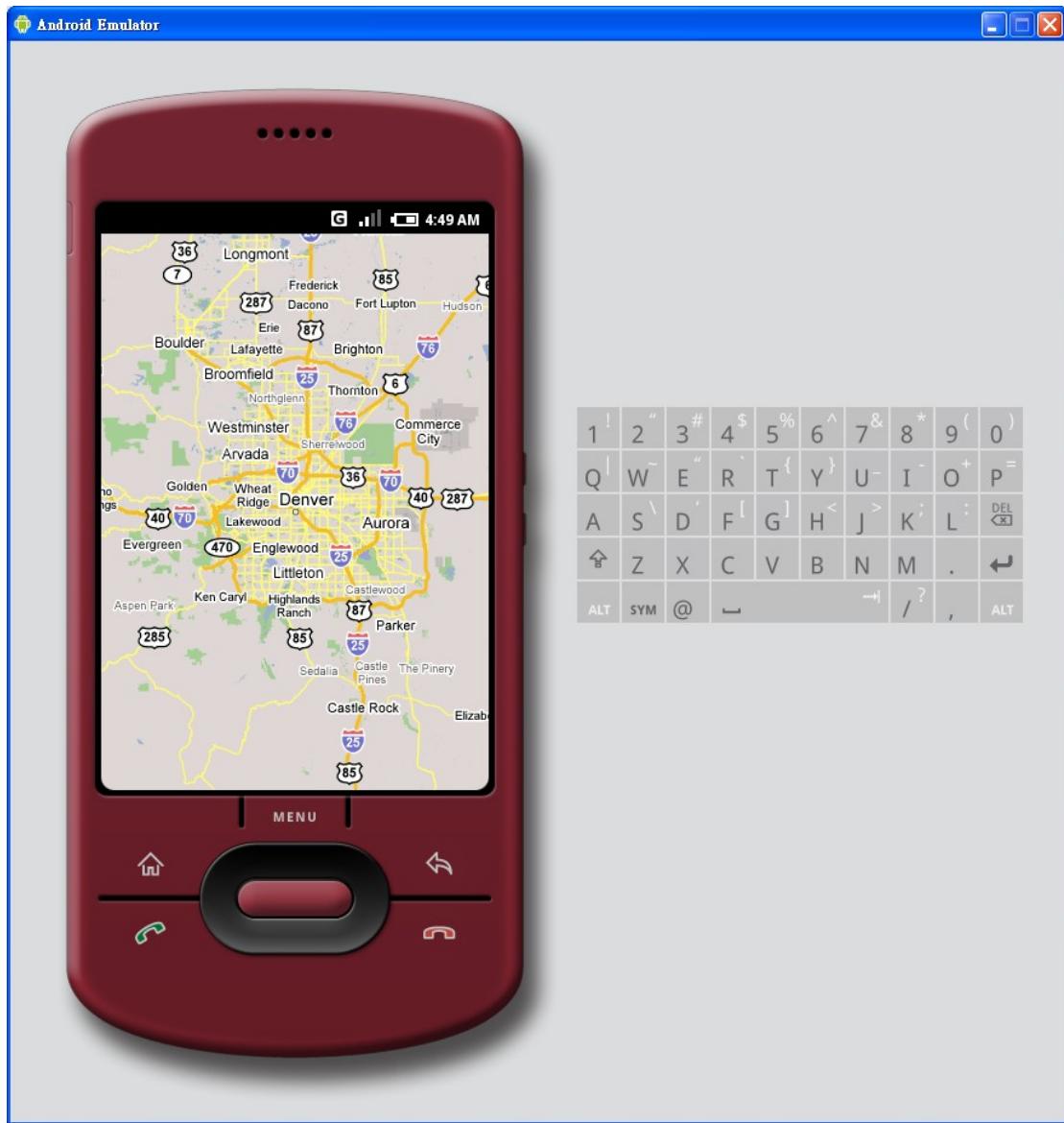


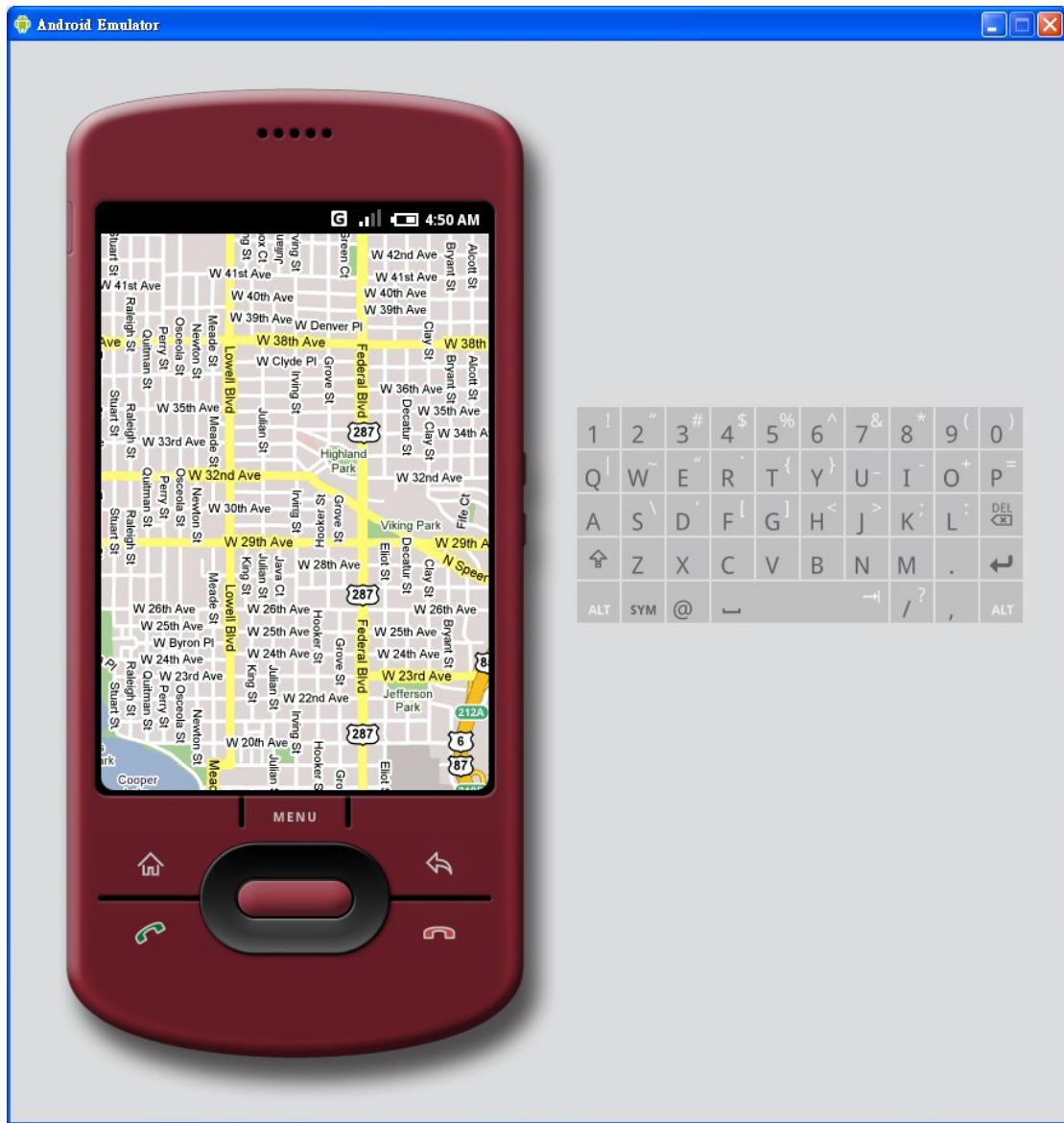














以下、我們將採用預設的 QVGA-L 版本進行操作與使用說明，當你已經安裝 Android 環境後，tools 資料夾下就會具有許多開發 Android 所需要的工具，以筆者的環境而言，我將 Android 環境安裝在 D:\Android\ sdk 底下，因此、模擬器 emulator.exe 的位置是位於 D:\Android\ sdk\ tools\ 底下，如下圖所示：

