

發佈應用程式

Android 系統架構使用資料夾觀念管理各國語言，讓開發者不費吹灰之力就可建立具有各國語言的應用軟體。**Android** 系統提供了「**support-screens**」宣告，能在絕大部分不同解析度螢幕上正常顯示。

應用程式完成後可以發佈到實機上測試，甚至上傳到 **Android** 市集網站，是每個開發都者應該要學習的重點。

學習重點

- 開發多國語言應用程式
- 開發支援各種螢幕解析度
- 安裝應用程式到實機
- 應用程式簽署私人金鑰
- 發佈應用程式到 **Android** 市集

16





16.1 支援多國語系及螢幕模式

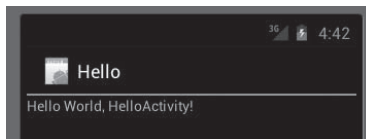
網際網路興盛縮小了世界的距離，Android 系統設計之初就放眼全球，系統架構使用資料夾觀念管理各國語言，讓開發者不費吹灰之力就可建立具有各國語言的應用軟體。Android 程式製作多國語系的方法，是將顯示的訊息文字獨立出來，避免與程式碼混雜在一起，然後以資料夾放置在不同的地方，就可輕易的根據不同語言顯示對應的訊息文字。



在開發上另一個問題，是設備的螢幕大小。每個月在市場上出現的 Android 的設備如過江之鯽，各種尺寸的螢幕更是讓人眼花撩亂。開發者當然希望其撰寫的應用程式在各種款式手機上都能正確無誤的執行及顯示，如何避免因使用者不同解析度的手機造成混亂，是開發者要注意的課題。

16.1.1 抽取字串到 XML 檔

在本書 2.1.5 節執行新建立的 Hello 空專案時，執行結果為：



顯示的歡迎訊息從何而來？在 Eclipse 整合環境開啟 Hello 專案，於 <res/layout/main.xml> 版面配置檔中可見到系統為新專案自動建立一個 TextView 元件，並設定其顯示內容為「@string/hello」：

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
/>
```



16.1.3 開發多國語言應用程式

當開發者將字串抽離置於獨立檔案後，就完成了製作多國語言應用程式的準備工作了！Android 系統會根據手機的語系設定，自動從 <res> 資料夾選擇對應的字串資源檔來使用。所以要建立多國語言應用程式非常容易，只要將原來字串資源檔複製後，將其字串內容改為需要的語言後，再置於指定的資料中就完成了！

切換語系

要觀看多國語言效果，可將模擬器切換到不同語系，就可顯示該語系的顯示結果了！切換語系的操作方法為：

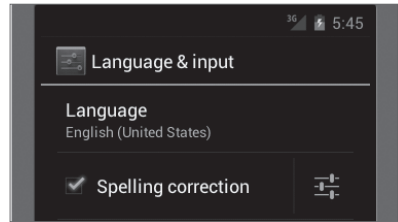
在模擬器功能區按 **MENU** 鈕，再按下方 **系統設定** 鈕。



在 **設定** 面板中選按 **語言與輸入裝置**，再在 **語言與輸入裝置** 面板中按 **語言**。



在 **地區設定** 面板中點選要設定的語言，此處選 **English (United States)**，模擬器就會以選擇的語言顯示。



語系資料夾名稱

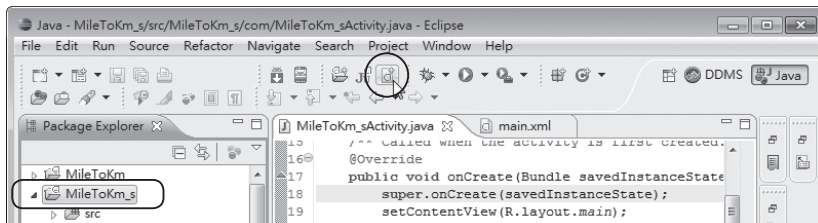
常用的語系在 <res> 中對應的資料夾名稱整理如下表：

資料夾名稱	語言
values-zh-rTW	繁體中文
values-zh-rCN	簡體中文
values-en	英文
values-ja	日文
values	無對應語系資料夾時使用

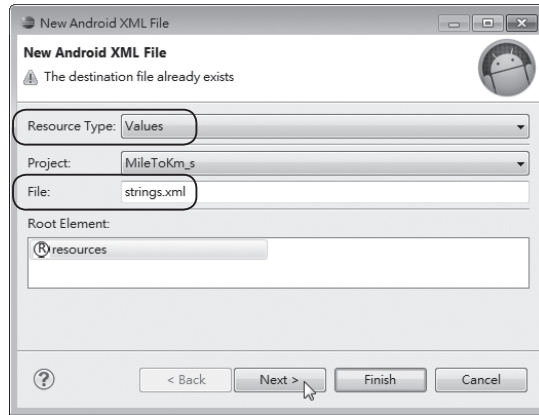
使用工具建立檔案

在 <res> 中新增多國語言所需的資料夾及檔案，通常使用複製的方法建立。如果對於這些複製、貼上等檔案操作不熟練，Eclipse 特別提供工具建立指定的資料夾及檔案，相當方便。操作方式為：

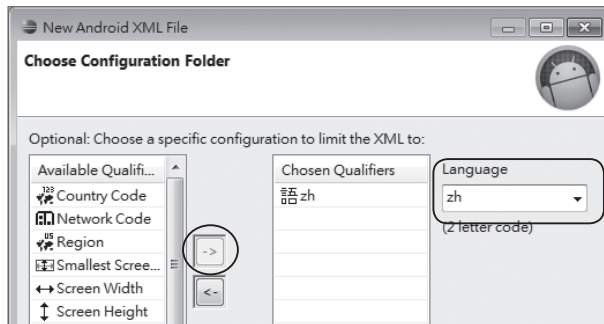
1. 於 Eclipse 整合環境的 Package Explorer 頁籤中點選要加入檔案的專案名稱，此處選 MileToKm_s，在工具列按  鈕。



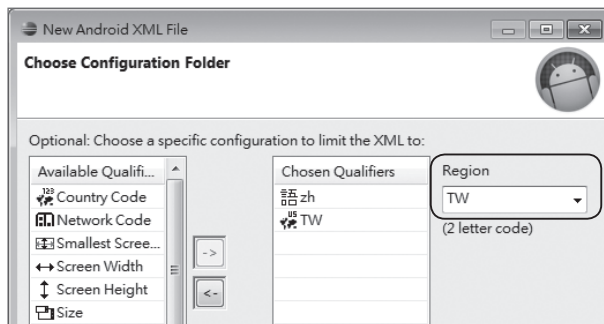
2. 於 New Android XML File 對話方塊中，Resource Type 欄位選取 Values，File 欄位輸入檔案名稱「strings.xml」，按 Next 鈕繼續。



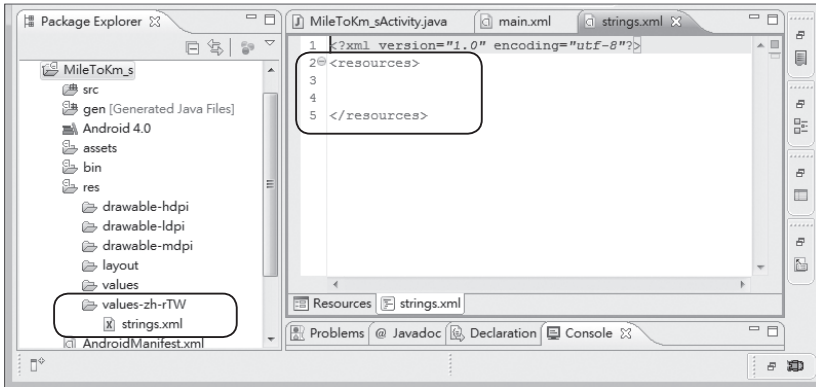
3. 在左方欄位選取 **Language** 後按 **->** 鈕，**Language** 欄位設定「zh」表示中文。



4. 在左方欄位選取 **Region** 後按 **->** 鈕，**Region** 欄位輸入「TW」表示繁體，按 **Finish** 鈕完成建立資料夾及檔案。

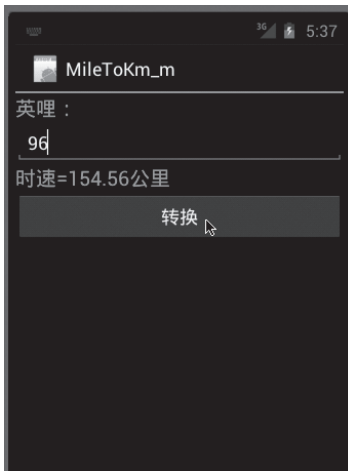


由 Package Explorer 可看到系統在 <res> 中已建立 <values-zh-rTW> 資料夾，內含 <strings.xml> 檔案，檔案內容已建立好 <resources> 標籤，開發者可直接加入字串程式碼。

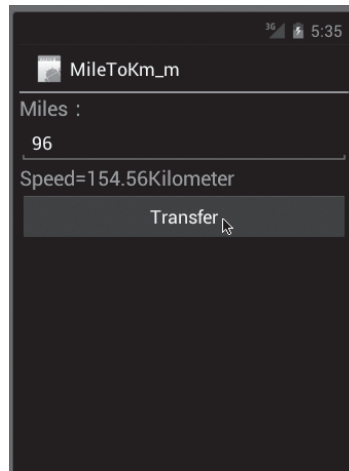


16.1.4 範例：計算大聯盟球速多國語言版

本範例延續前一範例，將顯示改為繁體中文、簡體中文及英文多國語言，若語系設定為其他語言則以繁體中文顯示。



▲ 簡體中文

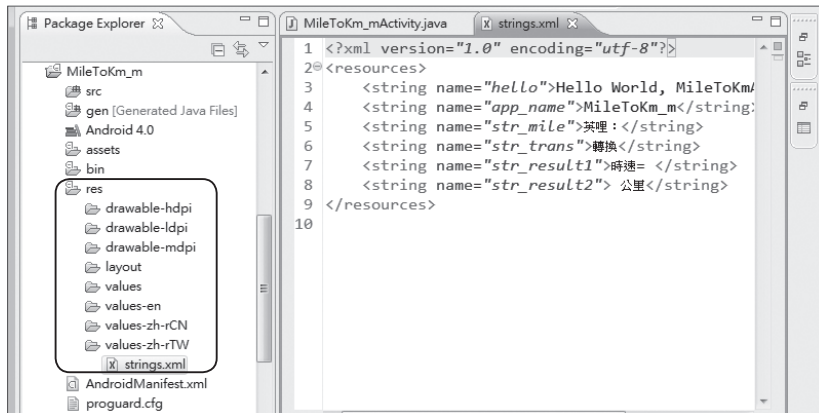


▲ 英文



匯入專案並以 Package Explorer 觀看資料夾結構

請匯入 <MileToKm_m> 專案，在 Package Explorer 觀看資料夾結構：



字串資源檔程式碼

預設 <res/values/strings.xml> 及繁體中文 <res/values-zh-rTW/strings.xml> 字串資源檔內容：

```
<MileToKm_m/res/values-zh-rTW/strings.xml>

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, MileToKmActivity!</string>
  <string name="app_name">MileToKm_m</string>
  <string name="str_mile">英里</string>
  <string name="str_trans">轉換</string>
  <string name="str_result1">時速 = </string>
  <string name="str_result2">公里</string>
</resources>
```

簡體中文 <res/values-zh-rCN/strings.xml> 字串資源檔內容：

```
<MileToKm_m/res/values-zh-rCN/strings.xml>

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, MileToKmActivity!</string>
```

```
<string name="app_name">MileToKm_m</string>
<string name="str_mile"> 英里 :</string>
<string name="str_trans"> 转换 </string>
<string name="str_result1"> 时速 = </string>
<string name="str_result2"> 公里 </string>
</resources>
```

英文 <res/values-en/strings.xml> 字串資源檔內容：

```
<MileToKm_m/res/values-en/strings.xml>
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, MileToKmActivity!</string>
  <string name="app_name">MileToKm_m</string>
  <string name="str_mile">Miles :</string>
  <string name="str_trans">Transfer</string>
  <string name="str_result1">Speed= </string>
  <string name="str_result2"> Kilometer</string>
</resources>
```

儲存專案後，請按下 **Ctrl + F11** 執行專案。您會發現在繁中、簡中及英文的語系下，程式都能正確以所屬的語系顯示一樣的內容。

16.1.5 支援各種螢幕解析度

可安裝 **Android** 系統的手機種類繁多，各種大大小小的螢幕解析度都有，開發者不太可能為各種解析度都撰寫適合其顯示的版面配置。**Android** 系統在 **1.6** 以後的版本提供了「**support-screens**」宣告，只需在 <**AndroidManifest.xml**> 檔中做些許設定，就能在絕大部分不同解析度螢幕上正常顯示。

開發者苦心撰寫的應用程式，當然希望能在所有 **Android** 系統的手機上執行，可在該應用程式的 <**AndroidManifest.xml**> 檔中加入下列程式碼：

```
<support-screens>
  android:smallScreens="true"
  android:normalScreens="true"
  android:largeScreens="true"
  android:xlargeScreens="true"
  android:anyDensity="true"
</support-screens>
```

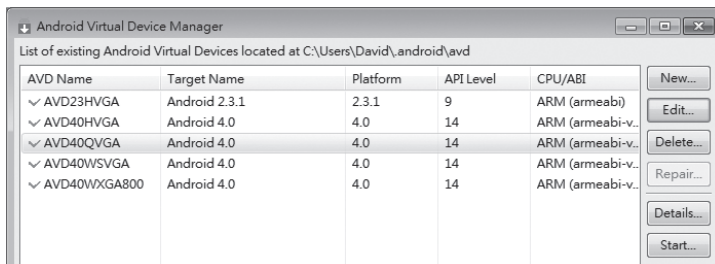
可別忘了加入啊！





使用者在 **Android Market** 搜尋可用的應用軟體時，系統會根據手機解析度顯示該解析度可使用的軟體，加入以上的設定，當應用程式發佈到 **Android Market** (發佈方法將在下一節說明) 後，無論使用者的手機螢幕是何種規格，都會顯示在搜尋結果中。

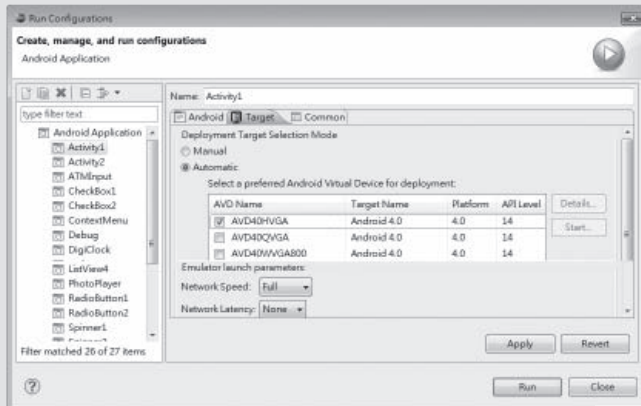
不過，開發者需留意，加入以上支援各種解析度的宣告，只是「宣稱」應用程式可以在各種解析度中顯示，但是否真的一定可以執行無誤，由於程式千變萬化，**Android** 系統無法保證。開發者應建立各種解析度的模擬器，在發佈之前一一測試，才能確保應用程式的正確性。



選擇測試的 AVD 模擬器

一旦建立的 AVD 模擬器多了，專案如何選擇測試使用的模擬器呢？

1. 請由 Eclipse 功能表的 **Run \ Run Configurations ..** 進入對話方塊。
2. 在左方選取要執行的專案後，在右方選按 **Target** 標籤，再核選 **Automatic** 時再選擇要使用的 AVD 模擬後按 **Apply** 鈕確定，或按 **Run** 鈕執行。



16.2 將作品發佈到 Android 市集

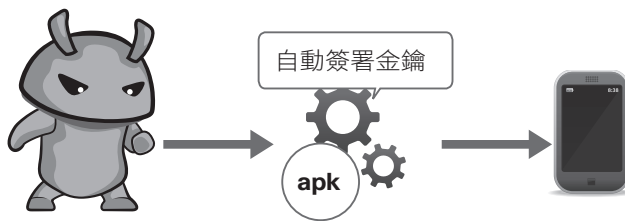
每一個應用程式都是發者花費不少時間精力的心血結晶，如果不能讓他人使用就太可惜了，若能利用撰寫的應用程式賺進些許「小朋友」就更更好了！Google 公司創建 Android 市集，供 Android 應用程式開發者將其成品發佈給所有人使用，開發者若自認應用程式具商業價值，可列為付費軟體與 Google 公司分帳，或以附掛廣告方式取得廣告費。

16.2.1 安裝應用程式到實機

開發視窗應用程式時，必須將開發專案編譯成附加檔名為「.exe」型式的執行檔，才能在視窗環境中執行；同樣的，Android 專案也必須編譯成附加檔名為「.apk」的安裝檔，才能在實體手機上安裝並執行。

Android 專案包含非常多資料夾及檔案，系統編譯專案的安裝檔時，會將專案中的版面配置、各種資源（圖形、字串等）、程式等全部包裝成一個安裝檔案，開發者只要將此安裝檔發佈出去，使用者就能輕易的安裝在自己的實機上。

Android 系統為了能保證應用程式的安全性，因此規定在產生安裝檔時，必須經過「簽署金鑰」的過程。前面章節在開發時，為了測試應用程式執行結果，會先編譯專案，然後在模擬器中執行，並沒有進行簽署金鑰過程，為什麼可以順利在模擬器中執行並觀看結果？那是因為 Eclipse 整合環境在編譯時會自動簽署一個名稱為「debug」的金鑰，好讓開發中的應用程式可以在模擬器上執行。

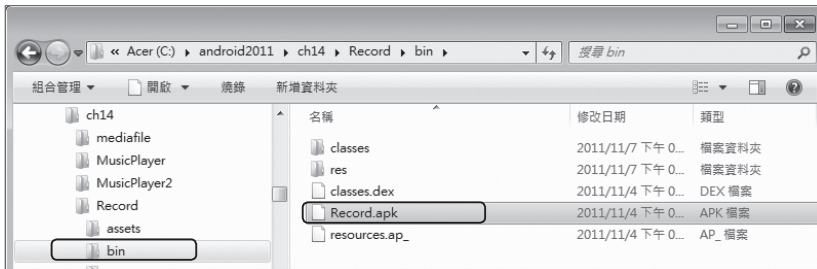


取得專案發佈的安裝檔

要將編譯完成的 apk 檔安裝於實機，首先要先把 apk 檔複製到手機的 SD 卡上，再使用 Android 應用程式安裝軟體來安裝。



此處以錄音程式：**Record** 專案為例，在模擬器中執行專案時，編譯完成的 **apk** 會儲存於專案根目錄的 **<bin>** 資料夾中，檔案名稱與專案名稱相同。連接手機與電腦後，複製 **<Record.apk>** 到 SD 卡上：



下載安裝軟體

Android 市場上有非常多 Android 應用程式安裝軟體，使用者可任意選擇一種使用。此處以下載安裝 **Apk Installer** 應用程式為例：

1. 於手機上應用程式桌面選按 **Market** 應用程式，出現 **Market** 網頁後按右上角搜尋 鈕 (手機必須在網路連線狀態)。



2. 搜尋字串輸入「**apk installer**」，按右方 鈕就會顯示搜尋結果，可以選取任一軟體，此處選按第一項 **Apk Installer**。

