



前言

過去手機的最大致命傷,就是應用程式的不足,雖然目前出現 了許多手機和 PDA 結合的產品,可是這些產品上的 PDA 在軟體上的 擴充性仍然是死的。我們可以任意使用 C/C++或是 Java 撰寫可以在 Palm OS、Windows CE、EPOC 等作業系統上執行的應用程式,但是卻 從未有過機會將我們寫好的程式下載到手機上執行(即使有些手機 的 PDA 也是使用 EPOC 作業系統,如 Ericsson R380)。但是從新的世 紀開始,這一切都要改觀了,陸陸續續有許多支援 Java 的手機即將 推出,我們終於可以一圓在手機上面撰寫程式的美夢。

ava

手機平台的多樣性與複雜性,和 PDA 比較起來可以說是有過之 而無不及,這等於是在提醒大家 -又是 Java 派上用場的時候了。我 們都知道在瀏覽器上執行的 Java 程式叫做 Applet,在 Palm OS 上執 行的 Java 程式叫做 Spotlet,然而在手機上執行的應用程式我們稱 做 MIDlet,本章將開始您的手機程式開發之旅。

### 下載開發工具 ▼

目前市面上有許多支援 MIDIet 的整合開發工具。但是筆者還是 堅持,在使用這些方便的開發工具之前,要先學著使用最基本的工 具來撰寫程式,這樣才能了解程式從開發到運作的細部過程。因此 在本章中我們只討論如何使用 Sun 所提供的基本工具 - MIDP 參考實 作來開發程式。

首先,請開啟您的瀏覽器,並連線到 http://www.javasoft.com /products/midp/

在網頁之中尋找一個叫做 Reference Implementation 的地方, 如圖1所示:

J2ME Wireless Toolkit	5 The <u>J2ME Wireless Toolkit</u> is a set of tools that provides Java technology developers with the emulation environment, documentation and examples needed to develop MIDP compliant applications targeted at mobile information devices such as cellular phones and two way pagers.	
MIDP Expert Group	Find out who are the members of the $\underline{\text{MIDP expert group}}.$	
Final Specification	Download the final specification of the MID profile.	
Reference Implementation	A reference implementation of the MID profile for the Microsoft Windows environment is available for <u>download</u>	
Technical Compatibility Kit	The Technical Compatibility Kit (TCK) for the MID profile can be licensed from Sun. For more information about licensing terms, please contact your <u>local sales representative.</u>	
See MIDP in Action	View some <u>sample applications</u>	
White Paper	Applications for Mobile Information Devices. Helpful Hints for	Java Technology

圖 1

接著,請按下名為 donload 的超連結,這個超連結會連到 MIDP 參考實作的下載網頁,但是在連線之前,需要您先輸入使用者 ID 和 密碼。如圖2所示:

2		
J	Java 手機程式設計入門	
Java		
	If this is your first visit,	
	please complete the registration form by clicking the "Register" button. You only have to register once. Whenever you come back to the Download Center, just enter your Personal ID and Password in the "Sign On" section below. We respect your privacy. Please see the top of the registration form for details.	
	Register	
	Sign On If you have registered before log in with your personal ID and password:	
	Personal ID Required Password Required Log In	

圖 2

tonio Road, Palo Alto, CA 94303 USA. All rights reserve

Copyright 1994-2000 Sun Microsystems, Inc Terms of Use. <u>Privacy Policy</u>. <u>Feedback</u>

如果您已經是會員,請直接輸入您的使用者 ID 和密碼即可進入 版權網頁。如果您並非會員,請先按下上面的 Register 按鈕註冊成 為會員,以取得使用者 ID 和密碼 (完全免費)。版權網頁如圖 3。

Original Contributor desires to license the Technology to a large community to facilitate research, innovation and product development while maintaining compatibility of such products with the Technology as delivered by Original Contributor; and Original Contributor desires to license certain Sun Trademarks for the purpose of branding products that are compatible with the relevant Technology delivered by Original Contributor; and
Please Choose:
C Accept C Decline
Continue

圖 3

3-4

F

請將畫面捲動到網頁最後的部分。先點選 Accept,然後再按下 Continue 按鈕以進入下載網頁,如圖 4。

Download Area	
<ol> <li>Choose your preferred download location and click "US West US Central US East US East US East Change</li> <li>Chick on the link below to navigate the download area a host that is not recognized to be in an <u>authorized co</u></li> </ol>	Change" to download from a different server. (optional) (NOTE: Link will not appear if you are downloading from unity).
<u>Mobile Information Device Profile</u>	Last Modified: 117 day(s) ago
Technical Support & Installation	
Mobile Information • Email: <u>midp-comme</u> Device Profile, 1.0	nts@sun.com
Customer Service	

圖 4

您可以先選擇離您最近的下載點,然後再按下 Mobile Information Device Profile 打開下載目錄,如圖 5。

Download Area
<ol> <li>Choose your preferred download location and click "Change" to download from a different server. (optional)</li></ol>
Mobile Information Device Profile
Technical Support & Installation
Mobile Information • Email: <u>midp-comments@sun.com</u> Device Profile, 1.0
Customer Service
<b>a</b>
圖 5





最後,請點選畫面上名為 j2me\_midp-1\_0-fcs-bin-b10-win-15\_sep \_2000.zip 的檔案即可下載 Sun 官方所提供的 MIDP 參考實作。

 $\overline{\mathbf{n}}$ 

# MIDP 參考實作 ▼

解開 MIDP 參考實作之後,您會看到許多檔案及目錄,底下我們 分別説明這些目錄的功用。

### 注意

此後我們皆假設您將 MIDP 參考實作解壓縮到 d:\底下,所以您的 MIDP 參考實作應該放置在 d:\ midp-fcs 目錄之中。

midp-fcs



build





Sun Microsystems 的網站上所提供的 MIDP 參考實作只有 Windows 版本與 Solaris 版本。喜歡 Linux 的朋友可能就會失望了。 不過已經有熱心的朋友已經將它移植到 Linux 上,您可以到 http://www.johndbeatty.com/j2me.html 找到 J2ME-CLDC-MIDP 在 Linux 上的版本。

為了今後的方便起見,請先將 build\win32\tools 目錄底下的 preverify.exe 拷貝到 bin 目錄下。

MIDP 參考實作裡頭附上了許多範例應用程式,都已經被編譯成 類別檔,它們都被放置在 classes\example 目錄底下,如果您對它 們有興趣,可以試著執行他們。要執行這些範例的方法如下:



- 1. 請先將切換到 src\example 目錄底下。
- 2. 執行指令

```
d:\midp-fcs\bin\midp -classpath d:\midp-
fcs\classes\example;d:\midp-fcs\classes -descriptor
run.jad
```



因為排版的關係,所以上述指令必須切割成數行。實際上的操 作中,所有的指令都應該處於同一行,而且中間有空格。這一點底 下將不再重複説明。

3. 執行畫面如下

【啓動畫面】



3-8

### 【應用程式選擇畫面】

單 色	彩色
Tutt Choose One: Sokoban ∰ Tickets Colors ∰ Stock ∰ Tiles ManuPalla <u>About</u>	Y     Imp       Choose One:     Imp       Imp     Sokoban       Imp     Tickets       Imp     Colors       Imp     Stock       Imp     Tiles       About

# 【Sokoban 應用程式】

單 色	彩色
Sokoban Level 0	Sokoban Level 0
<u>Undo</u> <u>Menu</u>	Undo Menu

# 【Tickets 應用程式】

單 色	彩色
♥I     Image: Constraint of the second sec	Y.III       Image: Alanis Morisette         Alanis Morisette         Auction Menu         Ticket Info         Make a Bid         Set an Alert         Settings





### 【Colors 應用程式】



### 【Stock 應用程式】



### 【Tiles 應用程式】

單  色	彩色
The second secon	
<u>Start Menu</u>	<u>Start Menu</u>

### 【ManyBalls 應用程式】



# 【Sampler 應用程式】



### 【Properties 應用程式】

單 色	彩色
♥II     IIID       System     Properties       Pree Memory =     131368       Total Memory =     300000       ™investition set     Exit	Y.III     Important       System     Properties       Pree Memory =     176804       Total Memory =     300000       Total Memory =     20000



# Java

# 【HttpTest 應用程式】

單色	彩色
약II ABC IIII ABC IIII ABC	乎II ABC () Http Error
javax.microeditio n.io.Connection NotFoundExcept ion	javax.microeditio n.io.Connection NotFoundExcept ion
About Menu	About Menu

# 【Pong 應用程式】

單色	彩色
♥II       ●       ●       Score: 000       Stop       Menu	▼II         Image: Score: 000           Stop         Menu

# 【StarCuriser 應用程式】

單色	彩色
YII     Import       Collisions: 06     ***       ***     ***       O     **       ***     **       Score: 27     Pause	YII     Image: Delise sins: 0.0%       Collis®ns: 0.0%     %       C     %       Score: 20        Pause     Menu

【SpaceInvaders 應用程式】



各位執行的時候,一定會發現您的模擬器上的畫面為單色,那 麼要如何讓模擬器變成彩色畫面呢?請在命令列之下輸入指令:

set SCREEN\_DEPTH=8

然後再重新執行

d:\midp-fcs\bin\midp -classpath d:\midpfcs\classes\example;d:\midp-fcs\classes -descriptor run.jad

即可看到彩色畫面。

如果您想要深入了解該範例,那麼您可以在 src\example 目錄 下找到其原始碼。

3-13

# MIDlet 開發流程 ▼

Java

MIDlet 的開發流程與 Spotlet 類似,但是多了一些步驟,如圖 6 所示:

0.00



圖 6

從圖中大家大致上可以看出從撰寫 MIDIet 到能夠在實體機器或 是模擬器上執行需要七個步驟:

1. 撰寫程式原始碼。

2. 編譯。

3. 預先審核 (Preverify)。

3-14 书

- 4. 撰寫描述檔。
- 5. 合成 JAR 檔。
- 6. 測試與除錯。
- 7. 將程式下載到實體機器上執行。或使用模擬器來執行。

如果您只是要單純在模擬器上測試,可以省略為六個步驟(少 了合成 JAR 檔的步驟):

- 1. 撰寫程式原始碼
- 2. 編譯
- 3. 預先審核 (Preverify)。
- 4. 撰寫描述檔。
- 5. 測試與除錯。
- 6. 使用模擬器來執行 MIDIet。

各位比較會感到奇怪的地方在預先審核、合成 JAR 檔以及撰寫描 述檔這幾個步驟,底下我們分別説明:預先審核--寫過 Spotlet 的朋 友可能覺得不奇怪,但是沒接觸過 J2ME 的人可能不瞭解需要這個步 驟的原因。因為在各位撰寫 Java Applet 或是 Java Application 的 時候從未做過這個步驟;要額外做合成 JAR 檔以及撰寫描述檔這幾個 步驟是為了配合 Java Application Manager。其實,這幾個步驟很容 易就可以完成,不像它的名稱看起來那麼地複雜。理由和用意我們都 已經在前面章節談過,如果您忘記了,請再回頭參考前面章節。接下 來我們就要一一解釋這幾個步驟的操作方式了。



### 注意

此後我們皆假設您將 MIDP 參考實作解壓縮到 d:\底下,所以您的 MIDP 參考實作應該放置在 d:\midp-fcs 目錄之中。

底下我們所編譯的類別名稱叫做 HelloMIDIet,而每一個步驟 的-classpath 選項會指向 d:\midp-fcs\classes;. 這意味著如果 d:\midp-fcs\classes 目錄之下也有名為 HelloMIDIet 的類別,將造 成之後令人感到疑惑的執行結果。很不巧地,d:\midp-fcs\classes 目錄中真的有一個名為 HelloMIDIet.class,所以在執行下面步驟 前,請先將這個檔案自d:\midp-fcs\classes 目錄中刪除。

步驟1:撰寫程式原始碼

請先撰寫您的第一個 MIDlet,我們將它取名為 HelloMIDlet. java,程式碼如下:

#### HelloMIDlet.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class HelloMIDlet extends MIDlet implements
CommandListener
{
    private Command exitCommand;
    private Display display;
    public HelloMIDlet()
    {
        display = Display.getDisplay(this);
        exitCommand = new Command("Exit", Command.SCREEN, 2);
    }
    public void startApp()
    {
```

```
TextBox t = new TextBox("Hello MIDlet", "Test
string", 256, 0);
      t.addCommand(exitCommand);
      t.setCommandListener(this);
      display.setCurrent(t);
   }
   public void pauseApp()
   {
   }
   public void destroyApp(boolean unconditional)
   ſ
   }
   public void commandAction(Command c, Displayable s)
   {
     if (c == exitCommand)
     {
         destroyApp(false);
         notifyDestroyed();
     }
   }
}
```

步驟 2:編譯

注

意

請撰寫一個名為 midjavac. bat 的批次檔,內容如下:

```
midjavac.bat
javac -g:none -classpath d:\midp-fcs\classes;. -
bootclasspath d:\midp-fcs\classes %1
```

因為排版的關係,將批次檔的內容用數行呈現,且兩行之 間有空白字元作為間隔。實際上的批次檔中整個指令必須在同 一行。



接著你可以在命令列輸入:

midjavac HelloMIDlet.java



如果程式沒有錯誤,那麼在目錄下就會產生 HelloMIDIet. class。

撰寫過 Java 程式的朋友或多或少都使用過-classpath 這個編 譯器選項,但是鮮少有人用過-bootclasspath 選項,如果您直 接輸入 javac 然後按下 Enter 來查看這個選項的意義,其實説 明也不太清楚,這裡我要補說説明:

在前面章節中曾説過:「『J2ME 所支援的核心類別函式庫是標準 核心類別函式庫的子集合』,充其量也只是名稱上為子集合而 已,這些類別函式庫子集合的實作卻不一定與標準的類別函式 庫相同」。當時我也曾提到,要編譯 J2ME 的程式之前都必須先 安裝 J2SE。如果我們只使用:

javac -g:none -classpath d:\midp-fcs\classes;. %1

來編譯我們的程式,那麼當程式中用到 MIDP 與 J2SE 之中同名 的類別時,編譯器預設會使用 J2SE 的版本,雖然這些類別的名 稱相同,可是實際上內部的實作卻完全不一樣。雖然使用了 J2SE 版本的類別來編譯我們的程式可以通過編譯,可是卻完全

不能在支援 MIDP 的手機上執行。所以我們必須利用bootclasspath 選項來改變 Java 編譯器的預設行為,讓編譯器 去使用 midp 版的類別函式庫(放在 d:\midp-fcs\classes 目錄 下)。

步驟3:預先審核

請撰寫一個名為 midver. bat 的批次檔,內容如下:

midver.bat
d:\midp-fcs\bin\preverify -classpath d:\midp-fcs\classes;. d %1 %2

#### 注意

因為排版的關係,將批次檔的內容用數行呈現,且兩行之 間有空白字元作為間隔。實際上的批次檔中整個指令必須在同 一行。

接著你可以在命令列輸入:

### midver HelloMIDlet HelloMIDlet

如果預先審核的過程沒有發生問題,就會在您的目錄下建立一個名為 HelloMIDIet 的子目錄,並將預先審核過的類別檔放到此目錄下。





### 注意

批次檔內的第一個命令列參數(%1),是為了讓 preverify 將預先審核的結果放到與命令列參數同名的目錄下。preverify 預設會將預先審核的結果放到 output 目錄下,但是為了不造成 混淆,所以我們利用-d 選項改變預設放置預先審核結果的目 錄。第二個參數則是您想要轉換的類別檔名稱。

我們的批次檔是將預先審核之後的類別檔放入與類別同名 的目錄之中。也就是説,如果您之前完成了預先審核,目錄下 會出現一個名為 HelloMIDIet 的子目錄,這會造成您下一次重 新編譯時產生如下錯誤訊息:

HelloMIDlet.java:4: HelloMIDlet clashes with package of same name

public class HelloMIDlet extends MIDlet implements CommandListener

۸ 1 error



這是因為編譯器發現目錄下有一個名為 HelloMIDIet 的目錄,對 Java 來說,意思就是有一個名為 HelloMIDIet 的 package。編譯器不准您的類別名稱和 package 名稱相同,所以

導致編譯錯誤。要解決這個問題,您可以在每次編譯前先刪掉 前一次預先審核所產生的目錄即可。

其實在此處您就可以開始測試您撰寫的 MIDlet。請換到經 過預先審核之類別檔的所在目錄,然後輸入:

d:\midp-fcs\bin\midp HelloMIDlet

D:\jdk1.3.0\_01\my\HelloMIDlet>d:\midp-fcs\bin\midp HelloMIDlet Classpath is not set. Defaulting to "."

如果執行無誤,那麼模擬器會出現,內容如下:



在此要特別注意的是,如果您不小心讓 MIDP 模擬器執行了沒有 經過預先審核的類別檔,螢幕上會出現錯誤訊息如下:

ALERT: Error verifying class HelloMIDlet

D:\jdk1.3.0\_01\my>d:\midp-fcs\bin\midp HelloMIDlet Classpath is not set. Defaulting to "." ALERT: Error verifying class HelloMIDlet



▶ 步驟4:撰寫描述檔

請撰寫一個名為 HelloMIDIet. jad 的描述檔,內容如下:

#### HelloMIDlet.jad

ava

```
MIDlet-Name: HelloWorld
MIDlet-Version: 1.0.0
MIDlet-Vendor: Sun Microsystems, Inc.
MIDlet-Description: Sample Hello World MIDlet
MIDlet-Info-URL: http://java.sun.com/j2me/
MIDlet-Jar-URL: http://127.0.0.1/HelloMIDlet.jar
MIDlet-Jar-Size: 1020
MicroEdition-Profile: MIDP-1.0
MicroEdition-Configuration: CLDC-1.0
MIDlet-1: HelloWorld,, HelloMIDlet
```

● 步驟 5:合成 JAR 檔

請撰寫一個名為 midjar. bat 的批次檔,內容如下:

#### midjar.bat

jar -cfm %1 %2 -C .\%3 .

### 注意

因為排版的關係,將批次檔的內容用數行呈現,且兩行之 間有空白字元作為間隔。實際上的批次檔中整個指令必須在同 一行。

接著你可以在命令列輸入:

midjar HelloMIDlet.jar HelloMIDlet.jad HelloMidlet

如果過程沒有發生問題,就會在您的目錄下建立一個名為 HelloMIDIet.jar的包裹檔。



HelloMIDIet. jar 的內容如下:

€∰Win	Zip - Hel	loMIDlet.	jar					
<u>F</u> ile	<u>A</u> ctions	Qotions	<u>H</u> e 10					
3 N	No.	候 Open	) Favorites	Add	C Extract	✓ View	CheckOut	🥮 Wizarc
Name					Туре \land		Modified	
♦९ He I Mar	lloM[D]e nifest.m	t.class  f			Java Sourc MF File	e	2001/1/27 2001/1/27	上午 09:4 上午 09:5
					]			►
Selecte	ed G files,	0 bytes		Total	2 files, 2KB			🔵 🔘 //.

請開啟 Manifest.mf,其內容應該如下:

🖉 MANIFEST.MF - 記事本	- O X		
檔案(E) 編輯(E) 格式(Q) 說明(H)			
Manifest-Version: 1.0	<b>A</b>		
MicroEdition-Configuration: CLDC-1.0			
MIDlet-Description: Sample Hello World MIDlet			
MIDlet-Version: 1.0.0			
Created-By: 1.3.0_01 (Sun Microsystems Inc.)			
MIDlet-Vendor: Sun Microsystems, Inc.			
MIDlet-Jar-URL: http://127.0.0.1/HelloMIDlet.jar			
MIDlet-1: HelloWorld,, HelloMIDlet			
MicroEdition-Profile: MIDP-1.0			
MIDlet-Jar-Size: 1020			
MIDlet-Name: HelloWorld			
MIDlet-Info-URL: http://java.sun.com/j2me/			
	-		





如果您的並非如此,可能是因為指令下達錯誤,請再重新檢 查。否則在之後步驟雖然不會發生錯誤,但是在將來配合網頁 伺服器來執行的時候將會出現錯誤。

ava

● 步驟6:測試與除錯

請撰寫一個名為 midjartest. bat 的批次檔,內容如下:

midjartest.bat
d:\midp-fcs\bin\midp -classpath d:\midp-fcs\classes;%1 descriptor %2

### 注意

因為排版的關係,將批次檔的內容用數行呈現,且兩行之 間有空白字元作為間隔。實際上的批次檔中整個指令必須在同 一行。

接著你可以在命令列輸入:

#### midjartest HelloMIDlet.jar HelloMidlet.jad

如果執行過程沒有錯誤,則會出現 MIDP 模擬器,內容如下:



按下圖上用圓圈框住的手機按鈕就可以執行程式,內容如下

3-24



您也可以隨時按下用矩形框住的手機按鈕離開程式。



如果之前您的類別檔沒有經過預先審核的動作,那麼在按下上 述圓形紅色方框執行程式時會出現錯誤訊息如下:

ALERT: Error verifying class HelloMIDlet

ALERT: Error verifying class HelloMIDlet

接著 MIDP 模擬器會立刻停止執行而跳出。





# 步驟7:將程式下載到實體機器上執行

當您拿到可以執行的 MIDIet 的手機時,應該至少有兩種方法可 以讓 MIDIet 下載到該手機上:

nvn

- 1. 透過網頁伺服器。
- 透過支援 MIDIet 的手機所提供的套件中與 PC 的連接線。有 關 MIDP 和網頁伺服器搭配的部分我們將在後面的章節位各 位解説。而利用連接線將 MIDIet 下載到手機上這個動作則 每家廠商各自有各自的解決方案,到時候還請您自行參考特 定手機的使用説明書。

最後,這些麻煩的步驟肯定讓您覺得很不方便,所以我們可以 再額外撰寫一個名為 midjaremu.bat 的批次檔,讓這些繁複的步驟 一次做完:

midjaremu.bat

```
call midjavac %1
call midver %2 %2
call midjar %3 %4 %2
call midjartest %3 %4
```

接著你可以在命令列輸入:

midjaremu HelloMIDlet.java HelloMIDlet HelloMIDlet.jar HelloMidlet.jad

D:\jdk1.3.0_01\my>midjaremu HelloMIDlet.java HelloMIDlet HelloMIDlet.jar HelloM Dlet.jad
D:\jdk1.3.0_01\my>call midjavac HelloMIDlet.java
D:\jdk1.3.0_01\my>javac -g:none -classpath d:\midp-fcs\classes;bootclasspath d:\midp-fcs\classes HelloMIDlet.java
D:\jdk1.3.0_01\my>call midver HelloMIDlet HelloMIDlet
D:\jdk1.3.0_01\my>d:\midp-fcs\bin\preverify -classpath d:\midp-fcs\classes;d HelloMIDlet HelloMIDlet
D:\jdk1.3.0_01\my>call midjar HelloMIDlet.jar HelloMIDlet.jad HelloMIDlet
D:\jdk1.3.0_01\my>jar -cfm HelloMIDlet.jar HelloMIDlet.jad -C .\HelloMIDlet .
D:\jdk1.3.0_01\my>call midjartest HelloMIDlet.jar HelloMIDlet.jad
D:\jdk1.3.0_01\ny>d:\midp-fcs\bin\midp -classpath d:\midp-fcs\classes;HelloMIDle t.jar -descriptor HelloMIDlet.jad

如果中間任何一個步驟沒有發生錯誤的話,這個批次檔可以幫 您將所有的步驟一次搞定。但是這種方便的工具也有其缺點,一般 常常發生的問題在您修改了程式之後,您的程式有誤,導致第一個 編譯步驟就發生錯誤,無法產生您預期中的新版類別檔,可是之後 的步驟卻仍然可以執行成功,這是因為前一版編譯成功的舊版類別 檔還存在目錄之下。這樣一來,您總是執行到舊版的程式,自然並 非您所預期的結果,所以請務必注意每一個步驟的輸出結果,任何 一個步驟發生錯誤都可能讓您執行到舊版程式。所以建議您每次重 新編譯時先刪去上一次所產生的所有檔案。

最後要額外告訴您的是,前面我曾提到,如果您只是單純想要 在 MIDP 模擬器上執行的話,您可以省略合成 JAR 檔的步驟。少了這 個步驟,很自然的上述的指令也必須經過修改:

請撰寫一個名為 midtest. bat 的批次檔,內容如下:



#### midtest.bat

nvn

d:\midp-fcs\bin\midp -classpath
d:\midp-fcs\classes;.\%1 -descriptor %2

### 注意

因為排版的關係,將批次檔的內容用數行呈現,且兩行之間有 空白字元作為間隔。實際上的批次檔中整個指令必須在同一行。

另外,-classpath 後面所接的參數是 d:\midp-fcs\classes;. \%1,沒有任何空白字元,絕對不是我打字錯誤喔!

接著你可以在命令列輸入:

#### midtest HelloMIDlet HelloMidlet.jad



如果執行過程沒有錯誤,一樣會出現 MIDP 模擬器。

當然,如果您要將這個繁複的過程自動化,您仍然可以撰寫一個批次檔來簡化您的工作,請額外撰寫一個名為 midemu.bat 的批次 檔,讓這些繁複的步驟一次做完:

#### midemu.bat

call midjavac %1
call midver %2 %2
call midtest %2 %3

接著你可以在命令列輸入:

midemu HelloMIDlet.java HelloMIDlet HelloMidlet.jad



如果中間任何一個步驟沒有發生錯誤的話,這個批次檔可以幫您將 所有的步驟一次搞定。就如同前面所説,方便的工具也有缺點,使 用時請務必小心觀看每一個步驟的執行結果。

# MIDP 參考實做對中文的支援 ▼

請嘗試修改 HelloMIDIet. java,內容如下:

### HelloMIDlet.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class HelloMIDlet extends MIDlet implements
CommandListener
{
    private Command exitCommand;
    private Display display;
    public HelloMIDlet()
    {
        display = Display.getDisplay(this);
        exitCommand = new Command("離開", Command.SCREEN, 2);
```

### Java

```
}
   public void startApp()
   {
      System.out.println("MIDlet 啓動");
      TextBox t = new TextBox("哈囉 MIDlet","測試 string", 256,
0);
      t.addCommand(exitCommand);
      t.setCommandListener(this);
      display.setCurrent(t);
   }
   public void pauseApp()
   {
   }
   public void destroyApp(boolean unconditional)
   {
   }
   public void commandAction(Command c, Displayable s)
   {
     if (c == exitCommand)
     {
         destroyApp(false);
         notifyDestroyed();
     }
   }
}
```

 $\overline{0}\overline{0}$ 

程式的執行結果如下所示:



3-30

大家可以發現中文可以正常地顯示,但是 System.out.println() 的輸出可就沒那麼好了,如下圖所示:

D:\jdk1.3.0_01\my>d:\midp-fcs\bin\midp	-classpath	d:\midp-fcs\classes;HelloMIDle	
t.jar -descriptor HelloMIDlet.jad			
MIDlet??			-

您可以將 midjavac. bat 的內容修改如下:

midjavac.bat
javac -g:none -encoding ISO8859_1 -classpath d:\midp- fcs\classes;bootclasspath d:\midp-fcs\classes %1

注意

因為排版的關係,將批次檔的內容用數行呈現,且兩行之間有空白字元作為間隔。實際上的批次檔中整個指令必須在同一行。

重新編譯之後結果會變成:

প্রনা। <sup>ABC</sup> ≪¢ÅoMIDIet	
ſú_Őstring	
L	Â÷¶}

螢幕上的中文全部變成亂碼,可是 System.out.println()的輸 出卻正常了:

3-31





nvn

因為有這麼奇怪的結果,所以在這裡建議大家如果要使用中 文,儘量只要在圖形使用者介面的地方使用中文即可,而除錯輸出 的部分請全部使用英文。

接下來一定有人希望也能在一開始選擇 MIDIet 執行的時候也能 輸出中文,讓應用程式的名稱用中文顯示:



請比對螢幕上的輸出以及前面 HelloMIDIet.jad 的內容,您會 發現決定應用程式名稱的地方就在 HelloMIDIet.jad 的最後一行的 部分,所以您可能會將 HelloMIDIet.jad 的內容修改如下:

### HelloMIDlet.jad

```
MIDlet-Name: Helloworld
MIDlet-Version: 1.0.0
MIDlet-Vendor: Sun Microsystems, Inc.
MIDlet-Description: Sample Hello World MIDlet
MIDlet-Info-URL: http://java.sun.com/j2me/
MIDlet-Jar-URL: http://127.0.0.1/HelloMIDlet.jar
MIDlet-Jar-Size: 1020
MicroEdition-Profile: MIDP-1.0
MicroEdition-Configuration: CLDC-1.0
MIDlet-1: 哈囉世界,,HelloMIDlet
```

您會發現啟動 MIDP 模擬器之後螢幕上會變成:



執行此程式螢幕上還會出現

Tatl 📼
(W can nor be
started using
class
null.
Exception:
java.lang.NullPoi
nterException
Done

同時命令列也會出現錯誤訊息如下:



要解決這個問題,我們必須運用一點小技巧,請將 HelloMIDIet. jad 的檔名改成 HelloMIDIet. jad\_old。然後在命令列輸入指令:

native2ascii -encoding Cp950 HelloMIDlet.jad\_old HelloMIDlet.jad



新產生的 HelloMIDIet. jad 如下所示:

HelloMIDlet.jad MIDlet-Name: HelloWorld MIDlet-Version: 1.0.0 MIDlet-Vendor: Sun Microsystems, Inc. MIDlet-Description: Sample Hello World MIDlet MIDlet-Info-URL: http://java.sun.com/j2me/ MIDlet-Jar-URL: http://127.0.0.1/HelloMIDlet.jar MIDlet-Jar-Size: 1020 MicroEdition-Profile: MIDP-1.0 MicroEdition-Configuration: CLDC-1.0 MIDlet-1: \u54c8\u56c9\u4e16\u754c,,HelloMIDlet

請重新執行模擬器,這次我們終於成功地讓選擇 MIDlet 的畫面 出現中文了:

 $\overline{0}$ 



之所以選擇 MIDIet 的畫面有中文問題,是因為 MIDP 規格中規 定描述檔的內容一定要使用 Unicode 編碼,所以如果我們一定要將 應用程式的名稱用中文表示,就必需額外利用 native2ascii.exe 這 個工具,將描述檔中使用 Big5 編碼的文字轉換成中文以 Unicode 編 碼的描述檔。

# MIDP 模擬器的執行選項 ▼

MIDP 参考實做內附的 MIDP 模擬器的功能其實很多,本文用到 的其實只有少數幾個。表 1 我們將説明這些選項及其用途:

選項名稱	用法
-help 列印功能選項	midp -help
-version 列印版本	midp -version D:\jdk1.3.0_01\my>d:\midp-fcs\bin\midp -version Jersion: MIDP-1.0 b10 Configuration: CLDC-1.0
-debug 開啟除錯功能	midp -debug
-verbose 執行時印出執行細節	midp -verbose GC heap size: 300000 bytes Invoking static initializer Invoking static initializer

表1 MIDP 模擬器執行選項



Java

		1)	
$\checkmark$	UU	/	00

選項名稱	用  法
-classpath <路徑或.jar 檔>類別 檔的所在路徑或 JAR 檔	midp -classpath foo.jar;c:\;.
-descriptor <.jad 檔>使用描述檔 來啟動程式	midp -descriptor foo.jad

# MIDP 模擬器的執行環境 ▼

上述我們的程式都是在 MIDP 參考實做內附的 MIDP 模擬器中作 測試,根據標準實做內附的文件所説,這個內附模擬器在執行時可 以藉由一些環境變數來改變其特性,底下我們一一介紹這些特性。

環境變數名稱	型態	預設値	說明
CLASSPATH	串		MIDP 模擬器會根據此環境變數 中所指定的目錄依序搜尋類別 檔。預設的目錄為".",也就是 您的提示符號當時所在之目錄 位置。
DOUBLE_ BUFFERING	甲	true	這個選項對於使用低階圖形處 理功能的人特別有用。可以有 效消除撕裂(tearing)的情形。
ENCODING	字串	null	我們使用的繁體中文可以正常 地在模擬器中顯現出來,所以 不用在意此環境變數。

表 2 MIDP 執行時期環境變數

環境變數名稱	型態	預設値	說明
SCREEN_DEPTH	字串	2(即 4 色 灰階畫面)	2==>四色灰階 4==> 16 色 8==> 256 色
ENABLE_CLDC_PR0 TOCOLS	宇	null	如果設定了此環境變數,那麼 CLDC 所能使用的傳輸協定將能 夠在 MIDP 中利用 Connector. open()來使用。

至於設定環境變數的方法,會根據您所使用的作業系統而有所 不同。如果您是在 Windows 作業系統下,那麼您可以使用指令如:

#### set SCREEN\_DEPTH=8

就可以設定表 2 中所提到的環境變數。

随時您都可以使用指令:

#### echo %SCREEN\_DEPTH%

觀看環境變數目前的值。



如果您是在 Unix 環境(如 Solaris)之下,那麼設定環境變數 的方法則會根據您所使用的 shell 而有所不同,請自行參閱該 shell 的説明文件來設定環境變數。



總 結 ▼

走了那麼遠一段路,終於完成了我們的第一個 MIDlet。雖然稍 後我們會介紹許多不同的開發工具,但是,最後您還是會發現,在 解決中文問題的時候,參考實做仍是最有威力的開發工具。

請不要砍掉在本章過程之中所產生的 HelloMIDIet.jar 與 HelloMIDIet.jad,這兩個檔案我們將會在往後章節中使用。