

1

Java 2 Micro Edition 概論

技術改版的消息，對工程師來說，是利空，但是對書籍作者來說，是利多。

- ▼ 前言
- ▼ 支援 Java 的行動電話
- ▼ 為何 Java 在行動通訊裝置上大放異彩?
- ▼ 各種 Java 版本的定位
- ▼ JINI 技術
- ▼ 各種不同版本之 Java 程式的開發
- ▼ Java 版本的演進
- ▼ Java 2 Micro Edition 規格
- ▼ CLDC、CDC、以及他們所衍生出來的 Profile
- ▼ Personal Java 及支援 Personal Java 的行動電話
- ▼ 有關 iAppli 與支援 iAppli 的行動電話
- ▼ 有關 STK
- ▼ 總結
- ▼ Java 2 Micro Edition 參考書籍

前言



如果您曾經到 Java 2 Micro Edition 的官方網站 <http://www.javasoft.com/j2me> 上查詢有關 Java 2 Micro Edition 的資料，十之八九會被一大堆的技術名詞搞的一頭霧水。什麼 KVM、CVM，什麼 CLDC、CDC、KJAVA、MIDP，後面還冒出了 Personal Java、Embedded Java 以及 JES 等名詞。雖然名為 Java 的微小版本，可是它的世界可真是不小，讓初次接觸它的人滿肚子「見山不是山，見水不是水」的疑惑。

因此本章將捨棄了技術上的細節，希望帶大家從巨觀的角度來看待 Java 2 Micro Edition 的世界。希望本章可以讓大家體驗「見山是山，見水是水」，一切豁然開朗的感覺。

在大家正式認識 Java 2 Micro Edition 之前，我們先為各位介紹目前所有支援 Java 2 Micro Edition 的行動電話，我們可以發現不管是歐、美、日大廠的手機都開始大力支援 Java 2 Micro Edition，藉此介紹可以讓大家體認，為何了解 Java 2 Micro Edition，以及學習如何用 Java 撰寫手機上應用程式肯定是一個無可避免的趨勢。



支援 **Java** 的行動電話 ▼

Motorola i85s



Motorola i50sx





Motorola A6288/Accompli 008 (太極+)



LG P520

請參閱網站 <http://java.ez-i.co.kr/>



Samsung X-120

請參閱網站 <http://www.xce.co.kr/>



 Sharp J-SH07



 Siemens SL45i/6688i





Mitsubishi J-D05



 Toshiba J-T06



如果您需要更多支援 Java 2 Micro Edition 手機的資料，請參考網站：<http://www.javamobiles.com/>

為何 **Java** 在行動通訊裝置上大放異彩？ ▼

行動通訊的時代即將來臨，通訊相關行業變的前景可期，而除了達成行動通訊的主要工具－手機，其功能越來越強大之外，有更多的廠商相繼投入行動通訊裝置的生產與開發，其平台和作業系統的複雜度比起目前混亂的 PDA 有過之而無不及，除了 Symbian 針對手機推出 EPOC 作業系統之外，微軟在手機上的作業系統 Stinger 也馬上就要亮相，但是目前市面上的手機多以廠商自行開發的作業系統居多。

面對複雜紛亂的平台，一般使用者當然沒什麼影響，但是對於程式開發者來說，這麼多不同的程式發展平台，光看完頭就昏了。如果每個平台都有自己的程式寫法以及函式庫，那麼光是看上面這些平台至少就要學習五種以上程式的寫法。當然，只專精一種平台當然是很好的事情。可是程式設計師不禁要說：“如果我們寫出來的軟體可以在不經過修改原始碼的情況下就能夠在這些平台上執行，那不是更完美嗎？”對程式開發者來說，這樣的投資報酬率當然是最大的。

要在那麼多平台上開發程式，對程式設計師來說的確是一個很大的挑戰，如果要把所有的時間和精力放在軟體的可用性上，那麼相對地很多時候我們根本沒有那麼多時間撰寫各種平台的程式。要解決這個問題，一般來說程式設計師會選用一個可以跨平台的 Framework 來達成至少 source code level 的跨平台（例入 Qt 就能做到）。不過在本章中我們要介紹的則是更終極的解決方案－Java，



利用 Java 的“write once, run anywhere”特性，我們可以真正達到程式只要寫一次，拿到任何平台上都可以執行（當然前提是必須要該手機廠商提供符合規格的 Java Virtual Machine 才行）。

利用 Java 撰寫手機上的程式當然有其缺點，最廣為人知的可能就是執行效率的問題，Java 在執行速度這個議題上一直讓人詬病。不過筆者認為，隨著技術的發達，將會有更快更省電的嵌入系統專用 CPU 出現，JavaOne 中也傳出將針對 J2ME 開發專用晶片，讓 Java 在手機上的程式可以跑的更快，因此效率上的問題其實是可以忽略的。更何況，當 Sun 在設計 J2ME 的時候，也用了很多方式企圖加快 Java 在手機上的執行速度（例如預先審核，將在稍後有詳細的說明）。

如果你有 Java 的基礎，想開始撰寫 Java 的手機程式，可以直接進入第二章「Java 程式設計簡介」，等有空再來看過以下的 J2ME 背景資料。

各種 **Java** 版本的定位 ▼

接下來，我們必須先對 Java 2 Micro Edition 在整個 Java 技術之中的定位做個了解。

使用 Java 撰寫過程式的人都知道，Java 的函式庫與一般大家所知道的函式庫有些不同：一般我們在撰寫 C 程式的時候，會使用 C 標準函式庫；使用 C++撰寫程式的時候，會使用 C++標準函式庫；而使用 Java 撰寫程式的時候，則使用類別函式庫。之所以叫做類別函式庫，是因為所有的函式被分門別類地歸類在不同的類別之下，相較起來，傳統標準函式庫的組成結構就顯的有點鬆散。



Java 規格之中有一組所謂的核心類別函式庫 (Core Class，即 `java.*`)，在核心類別之外還有所謂的擴充類別函式庫 (Extended Class，即 `javax.*`)。根據對這兩種類別所支援的程度，Sun Microsystems 進而區分出四種不同的 Java 版本，如圖 1 所示：

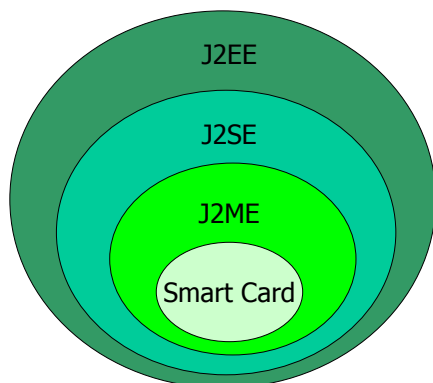


圖 1

圖 1 只是針對各種不同 Java 版本所支援之核心類別函式庫的範圍來做描述，並無法針對這些版本所支援的擴充類別函式庫做說明。另外，圖 1 也說明了這些版本所支援的 Java 基本型別 (primitive type，即 `boolean`、`byte`、`short`、`int`、`long`、`float`、`double` 這些型別) 之範圍。同心圓的越外面，所支援的核心類別函式庫就越完整。反之，位於同心圓內部的 Java 版本，所支援的就只是核心類別函式庫的子集合，而且越往裡面，所支援的核心類別函式庫子集合就越小。同理，越在同心圓的外部，所支援的 Java 基本型別就越完整，而越往同心圓內部，所支援的 Java 基本型別就越少。

請大家注意，在這裡我們說「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，這是因為即使是相同名稱的類別函式庫，為了對不同的平台予以最佳化（比方說執行速度和記憶體使用量）的理由，所以 Sun Microsystems 對類別函式庫子集合的實作做了翻修，因而導致類別函式庫雖然在名稱上相同，但是骨子裡卻不一樣的情形。

要理解圖 1，我們必須以 Java 2 Standard Edition (J2SE) 作為基準，這個版本實作了所有 Java 標準規格之中所定義的核心類別函式庫，也支援所有的 Java 基本型別。J2SE 定位在用戶端 (client-side) 程式的應用上。

從 J2SE 往外延伸，其外頭的同心圓為 Java 2 Enterprise Edition (J2EE)，此版本除了支援所有的標準核心類別函式庫之外，而且還增加了許多支援企業內部使用的擴充類別函式庫，比方說支援 Servlet/JSP 的 `javax.servlet.*` 類別函式庫、支援 Enterprise Java Bean 的 `javax.ejb.*` 類別函式庫。當然，J2EE 必定支援所有的 Java 基本型別。J2EE 定位在伺服器端 (server-side) 程式的應用上。

從 J2SE 向內看，首先會看到 Java 2 Micro Edition (J2ME)，它所支援的只有標準核心類別函式庫的子集合，比方說 J2ME CLDC 的規格之中，只支援 `java.lang.*`、`java.io.*`、以及 `java.util.*` 這些類別函式庫。J2ME 加入了一些支援嵌入式系統的擴充類別函式



庫，如 `javax.microedition.io.*` 類別函式庫。然而，此版本並不支援所有的 Java 基本型別，就標準 J2ME CLDC 的參考實作，也就是能在 PalmOS 上執行的 KVM (K Virtual Machine) 來說，它就不支援屬於浮點數 (float、double) 的 Java 基本型別。J2ME 定位在嵌入式系統的應用上。

同心圓的最裡面，還有一個 Java 的 Smart Card 版本，這個部分原本在 Java 的官方文件之中並沒有這樣定義，但是以筆者對所有 Java 版本的了解，將它畫在 J2ME 內部是非常合理的。因為 Smart Card 版本只支援 `java.lang.*` 這個核心類別函式庫，而且比起 J2ME 所支援的核心類別函式庫更少，但是它也有屬於自己的擴充類別函式庫，如 `javacard.*`、`javacardx.*` 這些類別函式庫。Smart Card 版本只支援 `boolean` 與 `byte` 這兩種 Java 基本型別。就如同其名稱，此版本定位在 Smart Card 的應用上。

在以上每一種 Java 版本之中，都有屬於它們自己的虛擬機器 (VM)，藉以達成「Write once, run anywhere。」的終極目標。在 Smart Card 上有 Card VM，負責執行下載到 Smart Card 上的 Card Applet；在 J2ME 的世界裡，其標準參考實作-KVM 及 CVM，KVM 用來執行下載至較低階（硬體功能不強）的嵌入式裝置上的 Spotlet 或 MIDlet，而 CVM 用來執行較高階（硬體功能相對較強）的簽入式裝置上的 Java 程式，；在 J2SE 與 J2EE 之中，有 Classic VM 與 HotSpot VM 負責執行 Java Applet、Java Servlet 或 Java Application。HotSpot VM 是新一代的 Java 虛擬機器，相較之下 Classic VM 則是傳統的 Java 虛擬機器。根據 Sun Microsystems 的



說法，HotSpot VM 執行 Java 程式的效率比起 Classic VM 有所提昇。如果您的電腦上裝有 Java 2 SDK 1.3.1，您可以在：

JDK 安裝目錄\jre\bin\classic

JDK 安裝目錄\jre\bin\hotspot

JDK 安裝目錄\jre\bin\server

這兩個目錄底下分別找到 jvm.dll，這兩個動態聯結函式庫就是我們所說的 Classic VM、Client HotSpot VM、以及 Server HotSpot VM。詳細的細節在此不多談，如果大家對 Hotspot VM 有興趣，可以參考

Java Platform Performance Strategies and Tactics
Addison Wesley 2000

這本書的附錄 B，以取得更多參考資料。

其實 HotSpot VM 還分成 client 版與 server 版。為何要區分成這兩種呢？其實主要是因為在用戶端執行的應用程式與伺服器上所執行的應用程式在基本需求上有所不同：用戶端應用程式比較注重畫面上的美觀、使用者介面的靈敏度；而伺服器上執行的應用程式則比較沒有畫面上的需求，但是比較注重對用戶端請求的回應。因此 HotSpot VM 的兩種版本就是為了針對這兩種不同的需求分別予以最佳化而產生的結果。因此所有 Java 版本與其所依據的虛擬機器所堆積起來的架構我們可以用圖 2 表示：



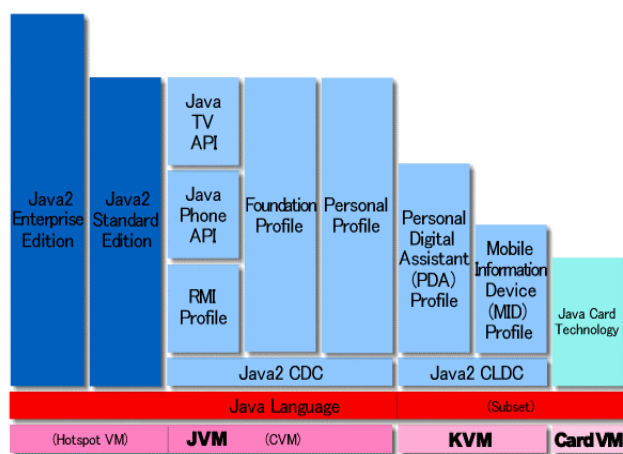


圖 2

以上所敘述的虛擬機器都只是 Sun Microsystems 根據 Java 虛擬機器規格的標準實作，實際上還有許多家公司都有自己所開發出來的虛擬機器，而且也符合 Java 虛擬機器的規格，比方說 IBM 的 J9 VM 就是 KVM 之外另一個支援 J2ME 的虛擬機器。所以大家千萬別以為 Java 虛擬機器只 Sun Microsystems 一家，別無分號。如果您對各家公司支援 J2ME 的虛擬機器有興趣，請參考表 1 所列的網站：

表 1

名 稱	URL
Sun KVM	http://java.sun.com/products/kvm/
Sun CVM	http://java.sun.com/products/cdc/cvm/
Kada 完整的 JVM	http://www.manthasoft.com/
Esmertec	http://www.esmertec.com/
IBM J9 VM	http://www.embedded.oti.com/palm/



JINI 技術 ▼

既然存在那麼多不同 Java 版本是一件事實，那麼，如果這些針對不同 Java 版本所開發的 Java 程式能夠彼此連結在一塊，相互連線，彼此分享各自的資源，豈不是一件很棒的事情嗎？這就是 JINI 之所以被創造出來的理由。各種不同的 Java 版本可以藉由 JINI 彼此聯繫，如圖 3 所示：

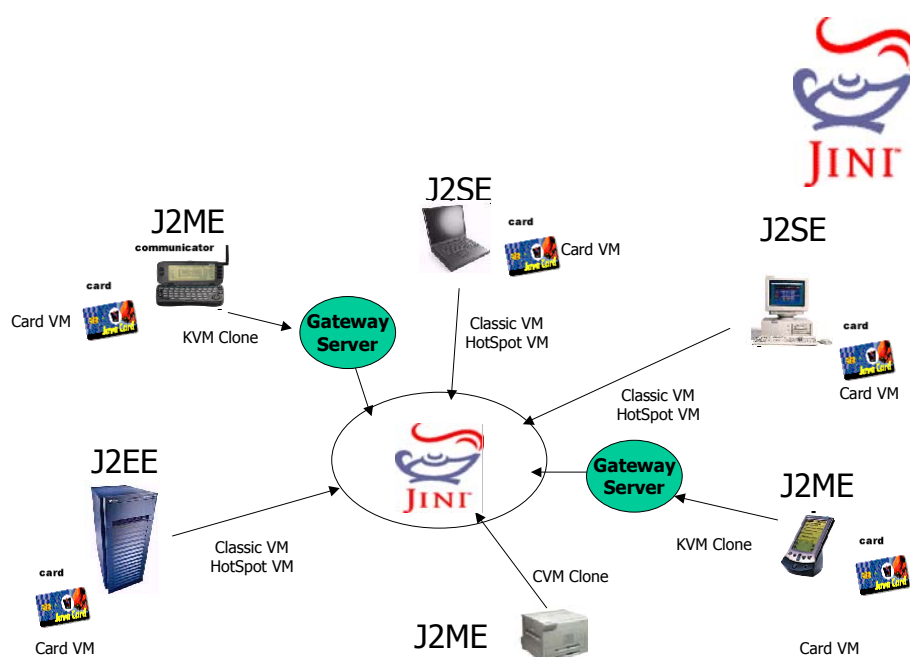


圖 3

有關 JINI 技術，在此不多著墨，因為市面上已經有許多相關書籍。不過在上圖之中要請大家注意的是，Smart Card 版本由於其特性使然，所以並沒有藉由 JINI 與其他 Java 版本的應用程式做溝



通，而只是在 JINI 所造成的廣大分散式環境之中作為認證之用。如果您要在 J2ME 上使用 JINI，並非無解，請您參考 Anhinga Project 首頁，網址為 <http://www.cs.rit.edu/~anhinga/>。

各種不同版本之 **Java** 程式的開發 ▼

如前面所說，各種不同的 Java 版本，在其支援的核心類別函式庫之完整性以及所支援的 Java 基本型別這兩件事情上都有所差異，但是對於程式設計師而言，這些版本的關係圖 4 所示：

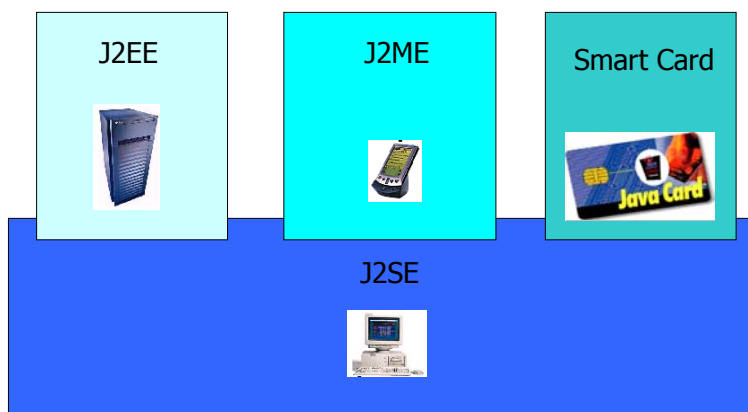


圖 4

也就是說，不管您開發的是企業所使用的 Java 程式、嵌入式裝置上執行的 Java 程式、瀏覽器上執行的 Applet，或是在 PC 上執行的應用程式。您都必須在您的電腦上先安裝 J2SE，然後再安裝各種版本的核心類別函式庫以及額外的擴充類別函式庫，如此才能成功地開發各種不同目的的 Java 程式。

J2SE 所提供的 Java 編譯器 (javac.exe) 可以幫助我們編譯各種不同平台上的 Java 程式，而 J2SE 所提供的 Java 虛擬機器 (java.exe) 則可以幫助我們在 PC 上先行測試這些程式執行結果的正確與否。

另外，Java 編譯器並不會幫您檢查您的程式是否符合各種平台上所支援的核心類別函式庫與 Java 基本型別。舉例來說，雖然我們在前面說過，Smart Card 版本並不支援 boolean、byte 以外的 Java 基本型別，而且該平台也只支援 java.lang.* 核心類別，可是當我們在撰寫 Smart Card 平台上的程式時，就算在程式碼裡頭用了 boolean 或 byte 以外的 Java 基本型別，或者使用了 java.lang.* 之外的其他核心類別，編譯器仍然可以照常幫您編譯出類別檔。這個時候大家一定開始產生疑惑--那麼這些程式如果放到 Smart Card 上頭執行的時候，出了問題怎麼辦？難道不會造成 Smart Card 上的虛擬機器執行時發生錯誤嗎？針對這個可能發生的潛在問題，Sun Microsystems 在各種不同版本的開發套件中，有些會內附檢查器 (checker) 或者預先審核器 (preverifier)，這兩個工具可以幫助您在將程式放到目標平台之前先做好檢查和預先審核的工作。

檢查器會幫您找出類別檔之中不合目標平台規格的部分，並提醒你這些地方可能無法在目標平台上執行。因此只要有檢查器的協助，您大致上可以確定您的程式可以符合目標平台的規定並順利執行。Java Card 的開發套件中就附有檢查器。



而某些平台的開發套件則附有預先審核器，預先審核器除了做檢查器做的工作之外，還有一項額外的工作，就是減輕目標平台上虛擬機器的負擔，要解釋預先審核器這個額外的工作，首先我們先看看傳統 Java 程式（Application、Servlet、Applet）的執行程序，如圖 5：

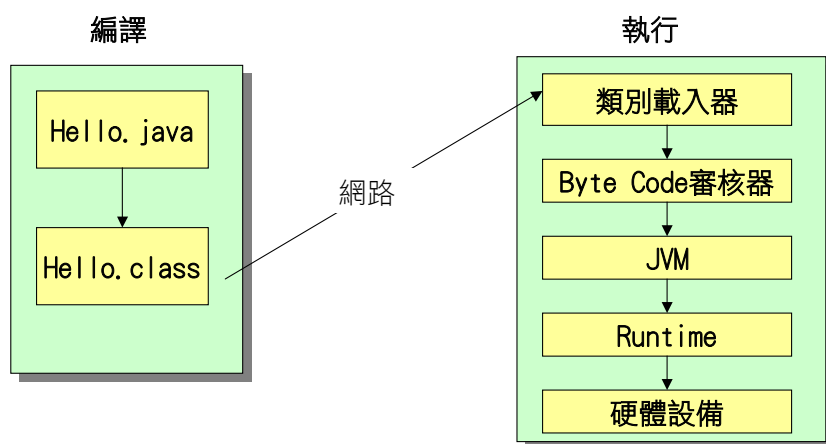


圖 5

在傳統的 Java 程式之中，為了安全上的考量，任何進入執行環境的類別檔（不管該類別檔是來自本機或是遠端機器），都必須先經過 Byte Code 審核器（Byte code verifier）的驗證，以防止程式在傳送途中遭到惡意的修改，而使得 Java 程式在執行時對系統有不良影響。經過審核之後，該類別檔才能開始被 Java 虛擬機器所執行。

如果這個審核的動作在一般的 PC 上執行，速度倒是還能夠接受，可是一旦放到如 Palm 或是手機這些 CPU 較慢、記憶體也比較少的機器上面就顯得十分吃力了。為了節省寶貴的 CPU 運算時間（既能省電又能夠加速程式執行），因此，在程式設計師產生能夠讓某些特定平台執行的類別檔之前，程式設計師必須先在 PC 上使用預先審核器做一些前置的審核工作，預先審核器會在類別檔之中加入一些特殊標記或符號。如此一來，當這些程式放到目標平台上執行時，就可以大幅減少在目標平台上做審核時的時間，藉而加速程式的啟動及執行速度。因此在 J2ME 之下的程式（Spotlet、MIDlet），其執行步驟變成圖 6 的樣子：

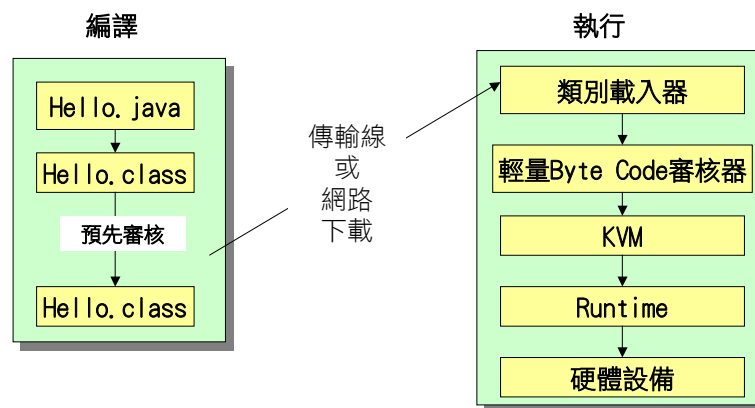


圖 6

從圖 6 之中大家可以看到，因為預先審核的關係，執行時 Byte Code 審核器的工作就變少了，也因此從程式載入到開始執行之間的時間因而縮短。CLDC 標準實作和 MIDP 參考實作之中就附有預先審核器。



Java 版本的演進 ▼

相信熟悉 Java 演進歷史的人或多或少都聽說過，Java 技術一開始並非就叫做 Java，而是叫做 OAK，而且最早的時候就是為了嵌入式系統而設計的一項產品。後來因為網際網路的發達，而 OAK 的諸多特性剛好又適合用在網路上（例如可移植性、編譯後程式碼很小），因為商標已被註冊的關係，因此 OAK 被改名成 Java，從此因緣際會地成了網路上的閃亮巨星，並隨著時間越來越成熟，也慢慢地產生了許多非原本預期中的相關運用。雖然 Java 已經被用到許多企業級軟體上，可是其實骨子裡面還是非常適合用在嵌入式系統之中。

Java 版本的歷史演進如圖 7 所示：

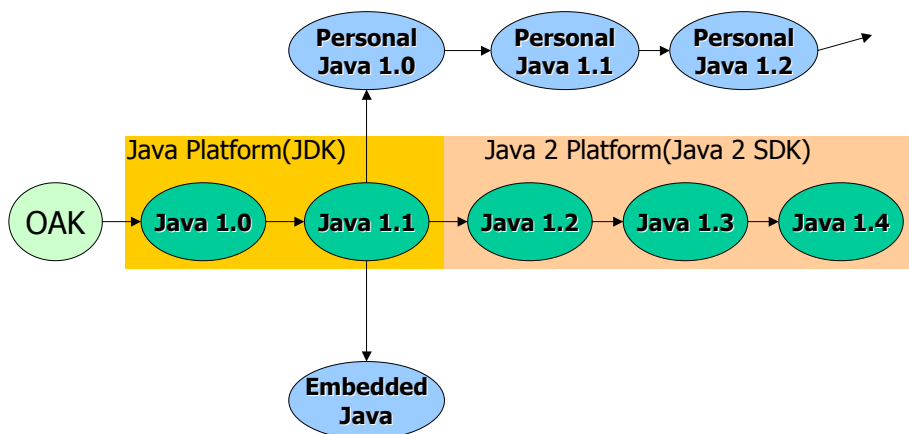


圖 7

從圖 7 中大家可以發現，雖然從 Java 1.0 發表之後，Java 就被廣泛地使用在桌上型應用程式以及 Applet 的開發上，但是，從 Java

1.1 開始，Java 又回到了它一開始的老路--也就是嵌入式系統方面的應用，在當時 Sun Microsystems 發表了 Embedded Java 與 Personal Java（也有人簡稱為 Pjava）這兩項規格。

其中 Embedded Java 是為了資源十分有限，而且沒有顯示設備的嵌入式裝置而設計；Personal Java 則是為了在能夠與網際網路連線、並擁有顯示系統（例如彩色 LCD）的消費性電子裝置而設計。

接著 Java 的版本演進到 Java 2，這時為了再明顯區分各種 Java 的應用，所以分割出了 J2EE、J2SE、以及 J2ME 三種版本。這三種版本的各種特性我們已經在前面已經詳細地描述，在此不再贅述。不過請大家記住，由於 Java 2 將 Java 的應用區分成三大塊，使得 Java 程式語言的發展不會再像 Java 1.1 時如樹枝狀般擴散出去，這麼一來有助於大家對 Java 各種應用的釐清，而不會造成今後越發展下去越不可收拾的混亂局面。

額外向大家一提的是，後來 Personal Java 發展到 1.2 版的時候，也採用了一些 Java 2 平台上與安全性有關的設計。

Java 2 Micro Edition 規格 ▼

J2ME 在設計其規格的時候，遵循著「對於各種不同的裝置而造出一個單一的開發系統是沒有意義的事」這個基本原則。於是 J2ME 先將所有的嵌入式裝置大體上區分為兩種：一種是運算功能有限、電力供應也有限的嵌入式裝置（比方說 PDA、手機）；另外一種則是運算能力相對較佳、並請在電力供應上相對比較充足的嵌入式裝置



(比方說冷氣機、電冰箱、電視機上盒 (set-top box))。因為這兩種型態的嵌入式裝置，所以 Java 引入了一個叫做 Configuration 的概念，然後把上述運算功能有限、電力有限的嵌入式裝置定義在 Connected Limited Device Configuration (CLDC) 規格之中；而另外一種裝置則規範為 Connected Device Configuration (CDC) 規格。也就是說，J2ME 把所有的嵌入式裝置利用 Configuration 的概念區隔成兩種抽象的型態。如果引入之前我們描述各種 Java 版本對於核心類別函式庫與擴充類別函式庫的概念，則 CLDC 與 CDC 的範圍應該如圖 8 所示：

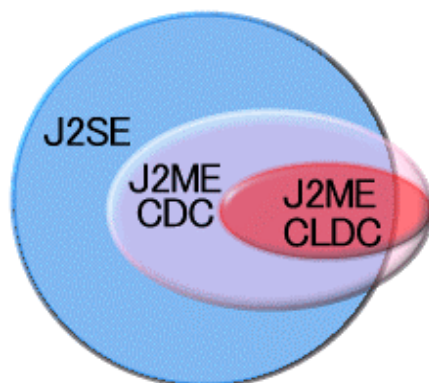


圖 8

其實在這裡大家可以把 Configuration 當作是 J2ME 對於兩種類型嵌入式裝置的規格，而這些規格之中定義了這些裝置至少要符合的運算能力、供電能力、記憶體大小等規範，同時也定了一組在這些裝置上執行的 Java 程式所能使用的類別函式庫、這些規範之中所定義的類別函式庫為 Java 標準核心類別函式庫的子集合以及與該型

態裝置特性相符的擴充類別函式庫。比方就 CLDC 的規範來說，可以支援的核心類別函式庫為 `java.lang.*`、`java.io.*`、`java.util.*`，而支援的擴充類別函式庫為 `java.microedition.io.*`。

區分出兩種主要的 Configuration 之後，J2ME 接著在定義出 Profile 的概念。Profile 是架構在 Configuration 之上的規格。之所以有 Profile 的概念，是為了要更明確地區分出各種嵌入式裝置上 Java 程式該如何開發以及它們應該具有哪些功能。因此 Profile 之中定義了與特定嵌入式裝置非常相關的擴充類別函式庫，而 Java 程式在各種嵌入式裝置上的使用者介面該如何呈現就是定義在 Profile 裡頭。Profile 之中所定義的擴充類別函式庫是根據底層 Configuration 內所定義的核心類別函式庫所建立。

嵌入式系統上的 Java 程式、Profile、Configuration、虛擬機器、作業系統、以及實體裝置之間的關係如圖 9 所示：

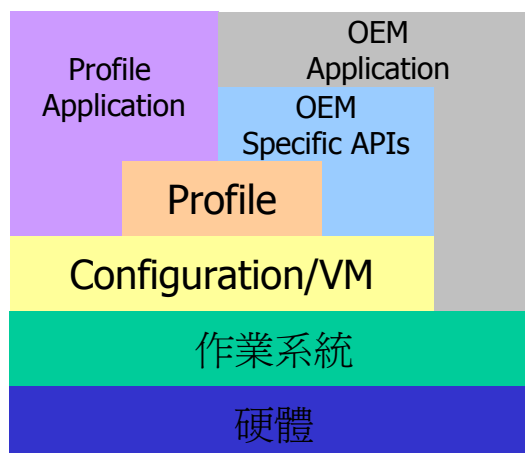


圖 9



從圖 9 大家一定可以發現，根據某個 Profile 規範所撰寫的 Java 程式除了可以直接呼叫 Profile 中定義的擴充類別函式庫，也可以直接叫用 Configuration 裡頭所定義的核心類別函式庫子集合與擴充類別函式庫。在這裡熟悉 Java 的朋友一定會問：「那麼在 J2SE 之中我們用來呼叫作業平台上之原生程式所使用的 Java Native Interface (JNI)，在 J2ME 之中是否仍然支援呢？從圖 9 看來似乎 J2ME 並不支援 JNI」。沒錯，當筆者翻閱了 CLDC 標準實作內附的 KVM Porting Guide 時，其第 10 章中有下面描述：

KVM does not support the Java Native Interface (JNI). Rather, the native code to be called from the virtual machine must be linked directly into the virtual machine, and must be called using the mechanisms described in this section.

Information for writing your own native functions for KVM is provided in Section 10.2, implementing native methods. ?

從這段敘述我們可以知道，除非您動手修改 CLDC 標準實作內附的 KVM 原始碼，並重新編譯和連結 KVM，否則我們無法充分利用 JNI 所帶來的便利（當然在這同時也會造成無法跨平台的問題）。

目前 J2ME 對於 Configuration 以及 Profile 所組成的架構大致如圖 10 所示：



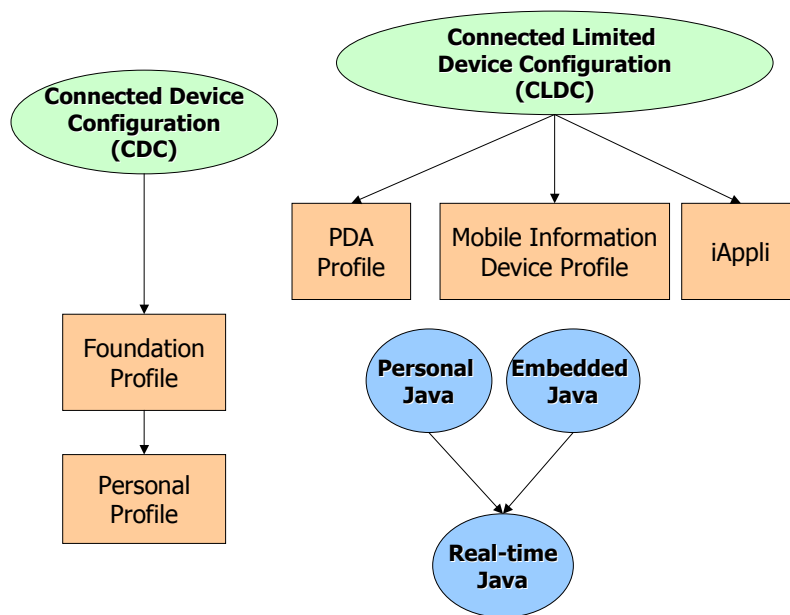


圖 10

從圖 10 可以看出，目前從 CLDC 所衍生出來的 Profile 有兩種，一種是 PDA Profile，大多數用在 PDA 上；而另外一種是 Mobile Information Device Profile（簡稱 MIDP），這個 Profile 是針對行動裝置所定義，比方說呼叫器（Pager）、行動電話等等，都是屬於行動裝置。

另外，我們可以發現 Embedded Java 與 Personal Java 似乎沒有包含在兩個 Configuration 任何一個底下，這看起來有點奇怪，不過他們還是歸類在 J2ME 的範疇之中，而且也衍生出了 Java 的即時版本（Real-time Specification for Java）。其實 Embedded Java 與 Personal Java 會慢慢被架構在 CDC 之上的 Profile 所取



代。拿 Personal Java 來說，其實它的規格與 Personal Profile 的規格是相容的。之所以目前還存有 Embedded Java 與 Personal Java 這兩個名詞的主要原因在於市場上已經有大量採用這兩種規格的實作產品，所以短期內 Personal Java 仍然會不斷演進，一時之間還無法立刻歸類到 J2ME 的特定 Profile 之下。

CLDC 以及他們所衍生出來的 Profile ▼

既然 Sun Microsystems 將所有的嵌入式裝置區分成 CLDC 與 CDC 兩種，接下來我們就這兩種 Configuration 做討論。

Connected Limited Device Configuration (CLDC) 就 Sun Microsystems 的官方文件所描述，是定義為「可以放在您的手掌上的裝置」，比方說 Palm 系列的 PDA 或是手機就是最好的例子。而 Connected Device Configuration (CDC) 根據 Sun Microsystems 的官方文件所述，定義為「可以插在牆壁上的裝置」，比方說電視機上盒 (set-top box) 就是一個例子。不管如何，這兩種 Configuration 之中定義的皆為這兩種型態的嵌入式裝置要執行 Java 程式所需要的最小配備需求。表 2 是 CLDC 與 CDC 各自的最小配備需求：

表 2

配備\ Configuration	CLDC	CDC
RAM	RAM 與 ROM 再加上快閃記憶體 (Flash Memory)要為 128K~ 512K。	大於等於 256K
ROM	RAM 與 ROM 加上快閃記憶體要為 128K~512K。	大於等於 512K
電源	通常是使用電池，所以電源有限。	不設限

在 Javasoft 的官方文件之中所架構出來的 J2ME 如圖 11 所示：

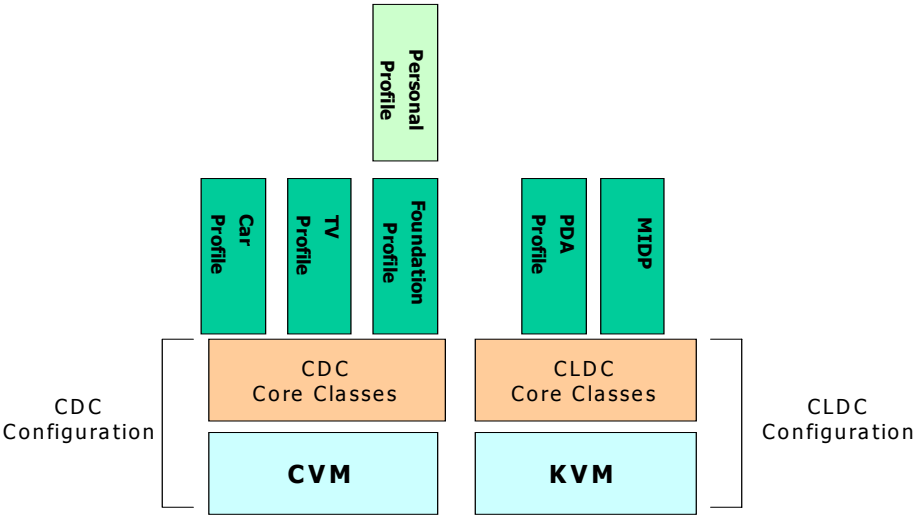


圖 11

在此大家也可以回頭參考圖 2，相信一定會更加清楚。



根據圖 11，我們來討論建構在 Configuration 之上的 Profile。在 Profile 之中也定義了特定種類嵌入式裝置的最小配備需求。既然 Profile 建構在 Configuration 之上，其意義就是說 Profile 之中所規範的配備需求不可能比 Configuration 還要低。同時，Profile 之中對於顯示功能、網路功能、以及耗電能力等相關需求將會比 Configuration 之中所的還要高。

表 3 是一些 Profile 的配備需求：

表 3

配備 \ Profile	Foundation Profile	Personal Profile	MIDP
RAM	至少 512K	至少 1 MB	RAM 與 ROM 至少 要為 512K
ROM	至少 1024K	至少 2.5 MB	RAM 與 ROM 至少 要為 512K
電源	不設限	不設限	通常是使用電池， 所以電源有限
網路連接能力	部分功能	部分功能	具有低頻寬的無線 通訊能力
其他	要有額外的 RAM 或 ROM 供應程式 執行	要有額外的 RAM 或 ROM 供應程式 執行	要有額外的 RAM 或 ROM 供應程式執 行並儲存資料

雖然官方文件之中繪出 J2ME 的美好遠景，其實這些規格還有待眾多參與規格建立之廠商的努力。在筆者撰寫本書的時候，我們所



能找到有關 Configuration 的規格之中，CDC 1.0 與 CLDC 1.0 的規格都已制定完成。而在 Profile 的規格方面，架設在 CLDC 之上的 Mobile Information Device Profile 1.0 的規格以及架設在 CDC 之上的 Foundation Profile 1.0 也都已底定，而其他如 Personal Profile、RMI Profile 的制定工作也在如火如荼的展開，相信很快就會出現在世人面前。如果您對這些已經制定完成或者正在草擬的規格有有興趣的話，請到 Java Community Process (JCP) 的網頁上觀察 Java Specification Request (JSR) 的後續進展。底下表 4 是 J2ME 相關的 JSP 之所在位置，表中深色的部分是本書討論的重點：

表 4

Java Community Process(JCP) http://jcp.org/
J2ME 相關 JSR 列表 http://jcp.org/jsr/tech/j2me.jsp
JSR #68 J2METM Platform Specification http://jcp.org/jsr/detail/68.jsp
JSR #134 Java Game Profile http://jcp.org/jsr/detail/134.jsp
JSR #135 Java Multimedia API http://jcp.org/jsr/detail/135.jsp



JSR #137 GAIN 3G Map Specification http://jcp.org/jsr/detail/137.jsp
JSR #30 J2ME Connected, Limited Device Configuration http://jcp.org/jsr/detail/30.jsp
JSR #36 J2METM Connected Device Configuration http://jcp.org/jsr/detail/36.jsp
JSR #37 Mobile Information Device Profile for the J2METM Platform http://jcp.org/jsr/detail/37.jsp
JSR #118 Mobile Information Device Next Generation http://jcp.org/jsr/detail/118.jsp
JSR #120 Wireless Telephony Communication APIs http://jcp.org/jsr/detail/120.jsp
JSR #82 Java API for Bluetooth http://jcp.org/jsr/detail/82.jsp
JSR #75 PDA Profile for J2METM http://jcp.org/jsr/detail/75.jsp
JSR #62 Personal Profile Specification http://jcp.org/jsr/detail/62.jsp



JSR #46 J2METM Foundation Profile http://jcp.org/jsr/detail/46.jsp
JSR #66 J2METM RMI Profile http://jcp.org/jsr/detail/66.jsp

雖然上數列表中的規格不少，但是已經底定的規格並不多，但是以目前來說，已經足以讓我們能夠撰寫 PDA（如 3COM PalmOS）和手機（如 Motorola、Nokia）上的應用程式。

各位在 CLDC 參考實作的內附文件之中，除了可以找到 CLDC 規格所訂定的核心類別函式庫子集與擴充類別函式庫之說明文件，也可以發現與 Palm OS 相關之擴充類別函式庫（com.sun.kjava.*）的說明文件。同理，我們除了可以在 MIDP 參考實作之中找到 CLDC 支援的核心類別函式庫子集與擴充類別函式庫之說明文件，同時也可以找到和 Mobile Information Device Profile (MIDP) 相關的擴充類別函式庫（javax.microedition.midlet.*、javax.microedition.lcdui.*、javax.microedition.rms.*）之說明文件。

Personal Java 及支援 Personal Java 的行動電話 ▼

在本章前面筆者曾經提到，Personal Java 的發展其實以一段時間了，無數的公司根據其規格生產實作品，而它所扮演的角色也在 J2ME 推出之後更形尷尬，所以接下來我們將別針對 Personal Java 做說明。



從前面的討論我們可以知道，Personal Java 的規格其實並沒有定義在 CLDC 或是 CDC 底下，雖然最後它將會被歸到 CDC 的 Personal Profile 之中，但是目前其規格還是繼續在演進之中。之所以有這種情況，是因為長期下來已經有許多圍繞著 Personal Java 的規格所實作出來的商業產品，造成尾大不掉的情形，所以 Personal Java 的規格短期還是會繼續演進。

前面在介紹 Java 版本的演進時，大家一定發現 Personal Java 的規格其實是從 Java 1.1 之中所分支出來，也因此 Personal Java 的規格是根據許多 Java 1.1 的規格而制定的，但是並非 Java 1.1 的全部規格都包含進來。Personal Java 特別適合用在具有豐富圖形顯示能力的消費性電子產品上面，於是我們可以發現 Sun Microsystems 網站上對於 Personal Java 的參考實作是建立在 Windows CE 上頭的，或許大家將 Windows CE 當作比較偏向於 PDA 的產品（它本來就是 PDA），可是 Windows CE 本身的確符合 Personal Java 規格之中所規格的目標平台之條件，比方說具有連接網際網路的能力、而且對於圖形的顯示能力非常強大（彩色 LCD）。另外一家公司 Symbian 所推出的作業系統 EPOC R5 以及 EPOC R6 上的 Java 版本也都是衍生自 Personal Java。

目前您可以在 Sun Microsystems 的網站上下載遵循 Personal Java 1.2 的參考實作之 3.1 版的原始碼以及 1.0 版的 Personal Java 執行環境 (Personal Java run-time environment)，不過比較可惜的是，1.0 版的 Personal Java 執行環境只支援 Windows CE 2.11 版，而且只支援 MIPS 與 SH3 這兩顆處理器。目前有很多執行



Windows CE 的 PDA 並非採用這兩種處理器，比方說有的機器是使用 StrongArm，如果剛好不巧您的 PDA 並非 MIPS 或 SH3，那麼您可能要到網路上尋找其他廠商所提供，可以在其他處理器上執行的 Personal Java 執行環境了。

就如同開發其他 PDA 程式一樣，您不一定要購買一台 PDA 來作測試，Sun Microsystems 網站上也提供了 Personal Java 的模擬器，讓您可以在 Windows 或 Solaris 作業系統之下測試您開發給 Personal Java 環境執行的應用程式。此模擬器名為 Personal Java Emulation Environment (PJEE)，您可以在 <http://java.sun.com/products/personaljava/pj-emulation.html> 下載 PJEE。另外，您必須使用 JDK 1.1.7 以上的版本來開發可以在 Personal Java 執行環境或 PJEE 上執行的應用程式。

或許在下載時您會注意到，目前您所能取得的參考實作分成兩種，一種叫做「Touchable」版本，另一種則是完整的版本。所謂的「Touchable」版本，意指“Personal Java 版本的最小實作”。其實 Sun Microsystems 網站上也建議我們，如果要開發 Personal Java 的應用程式，最好是以 Touchable 版本為基礎，除了很多商業版本的 Personal Java 實作是根據 Touchable 版本來開發的之外，也可以保證您的應用程式可以在所有的 Personal Java 實作上順利執行。

根據 Personal Java 所延伸出來的相關技術還有很多，表 5 列出其名稱以及參考網址，如果各位有興趣的話，可以自行在網路上參考這些資料。



目前市面上支援 Personal Java 的行動電話有：



NOKIA 9210



【註】NOKIA 9210 已經可以執行 MIDlet 程式，也就是說它已經能夠支援 MIDP 規格，您可以在 NOKIA 網站找到相關資料。



有關 iAppli 與支援 iAppli 的行動電話 ▼

iAppli 是由日本 NTT DoCoMo 自行開發的手機程式規格，它與 MIDP 一樣架設在 CLDC 之上，不過與 MIDP 並不相容。這意味著遵照 MIDP 規格所開發出來的程式無法拿到支援 iAppli 的手機上執行。這是因為 iAppli 的規格比 MIDP 的規格還要早制定完成的緣故。

雖然兩者並不相容，但是基本上兩者大同小異，底層都是架構在 CLDC 之上，除了使用的函式庫專屬於 NTT DoCoMo 之外，程式設計的概念差異不大。

目前支援 iAppli 的行動電話有：

NTT F503i 與 P503i





NTT FOMA N2001、P2101V、P2401



NTT N503i



 NTT D503i



 NTT SO503i





NTT P503Is



NTT D503iS

沒有相關圖片



有關 STK ▼

最後，我們向大家解釋一個或許大家在不久前曾經聽過的名詞--STK。

STK 又叫做 SIM Toolkit，全名為“Subscriber Identity Module Application Toolkit（用戶識別應用發展工具）”，是 Java 在 Smart Card 上的一種應用。這是透過行動電話公司與 SIM 卡製造商合作，在用戶識別卡的剩餘記憶體之中，直接燒錄能夠提供增值服務的程式，讓持有手機的使用者只要其 SIM 卡上已經有這些程式，就可以直接從手機目錄上點選他所需要的手機增值服務。

SIM 是用戶識別卡的簡稱，Tool、kit 都是工具的意思。顧名思義，就是如何應用 SIM 卡的工具。主要就是用來開發由 IC 晶片所構成的 SIM 卡上之應用程式。一旦 SIM 卡植入軟體，以便能接收來自於電信公司的很多增值服務，這些增值服務主要是大量而有用的資訊，如氣象報告、銀行理財、班機時刻表、股票資訊、電視節目表等等。

有了這些服務，就可以不必像過去，使用者必須要輸入一大堆的代號和密碼，才能瀏覽他想要觀看的。而要使 SIM 卡具有這種功能的方法，一是業者在卡片賣出時就設定好，另一方法是由電信公司從空中傳給用戶，而未來的趨勢是從空中直接傳過來（OTA-Over The Air）。不久之前幾家行動通訊業者所提供的行動銀行（Mobil Banking）業務就是 SIM Toolkit 應用的最佳範例。某銀行存款人只要拿起大哥大手機，就可以查詢銀行或任何金融卡的餘額、他可以



透過手機進行轉帳，十分方便。現在也有行動通訊業者提供電影節目表、電視時刻表、購票服務、計程車叫用服務、股市資訊傳送、氣象報告、投資理財資訊、班機班次時刻表等等。

但是由於 WAP 的興起，STK 所能提供的功能 WAP 都能夠做到，甚至更強大。對於開發者而言，WAP 網頁比起 STK 程式在開發上更為容易，也因此 STK 漸漸被 WAP 取代，慢慢地大家也比較沒聽過這個 STK 名詞。

總 結 ▼

看完了本章之後，您對整個 Java 2 Micro Edition 的架構已經有個大致上的認識嗎？在進入下一章之前，請務必弄清楚本章所提到的專有名詞。如果各位想對各種技術規格做更詳細的了解，請自行參考表 5 所列出的網頁。

表 5

名 稱	URL
Wireless Java	http://wireless.java.sun.com/
Consumer & Embedded Technologies	http://java.sun.com/products/consumer-embedded/
Consumer & Embedded Technology Center	http://developer.java.sun.com/developer/products/j2me/
Java 2 Micro Edition	http://www.javasoft.com/j2me/
JDC J2ME Tech Tips	http://developer.java.sun.com/developer/J2METechTips/



CLDC & KVM	http://www.javasoft.com/products/cldc/
名 稱	URL
CDC & CVM	http://java.sun.com/products/cdc/
C & E Center Wireless Technologies	http://developer.java.sun.com/developer/products/j2me/wireless.html
MIDP	http://www.javasoft.com/products/midp/
Java Wireless Developer Initiative Homepage	http://developer.java.sun.com/developer/products/wireless/
Embedded Java	http://www.javasoft.com/products/embeddedjava/
Personal Java	http://java.sun.com/products/personaljava/
JavaPhone	http://java.sun.com/products/javaphone/
Java Embedded Server	http://www.sun.com/software/embeddedserver/index.html
C & E Center Java Embedded Server™ Technology	http://developer.java.sun.com/developer/products/j2me/embedserv.html
JavaTV	http://www.javasoft.com/products/javatv/
C & E Center Digital Interactive Television	http://developer.java.sun.com/developer/products/j2me/tv.html
Java Card	http://java.sun.com/products/javacard/
C & E Center Java™ Card™ Technology	http://developer.java.sun.com/developer/products/j2me/smartcard.html
Motorola 官方 Java 網站	http://www.motorola.com/java/



JINI	http://www.sun.com/jini/
名 稱	URL
C & E Center Downloads & Specifications	http://developer.java.sun.com/developer/products/j2me/downloads.html
JGuru J2ME FAQ	http://www.jguru.com/jguru/faq/faqpage.jsp?name=J2ME
Spotless System Page	http://www.sun.com/research/spotless/

如果您對於特定技術有所問題，您可以參考表 6 所列出的線上討論區。

表 6

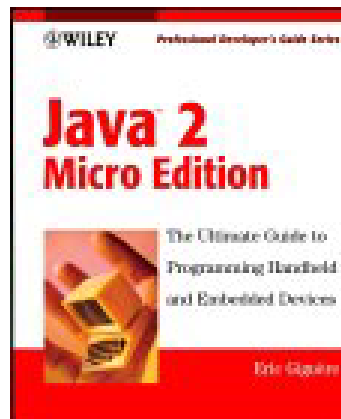
Programmer 深度論壇 J2ME 討論區 (中文)	http://forum.vclxx.org/
J2ME 俱樂部	http://j2me.uknows.net/
openbazaar.net J2ME 討論區 (中文)	bbs://linux.twbbs.org
PalmFab (中文)	http://palmfab.cx/indexForumData.php?FID=6
大陸 J2ME 討論群組 (簡體)	http://www.smiling.com.cn/group/homepage.cgi?group_id=16903
KVM-interest mailing list archives	archives/kvm-interest.html
KVM	http://forum.java.sun.com/list/discuss.sun.k.virtual.machine.kvm



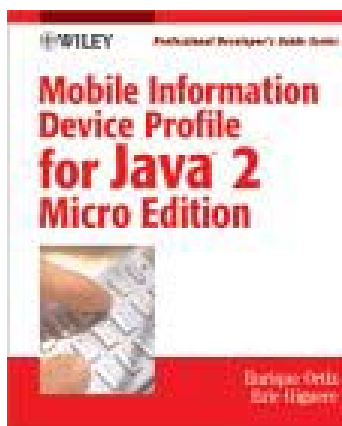
Java Embedded Server	http://forum.java.sun.com/list/discuss.sun.javaserver.products
Embedded Java	http://forum.java.sun.com/list/discuss.sun.embeddedjava
Personal Java	http://forum.java.sun.com/list/discuss.sun.personaljava
Java TV	http://forum.java.sun.com/list/discuss.sun.java.tv
Java Card	http://forum.java.sun.com/list/discuss.sun.java.card
JavaSM Wireless Connection Program	http://industry.java.sun.com/jwc/

Java 2 Micro Edition 參考書籍

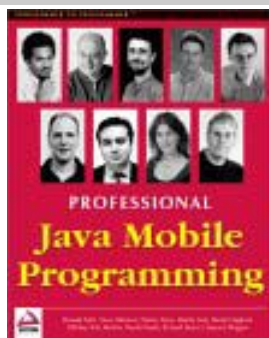
Java 2 Micro Edition : Professional Developer's Guide



http://www.amazon.com/exec/obidos/ASIN/0471390658/ref=pd_sbs_b_4/002-3586460-8574424

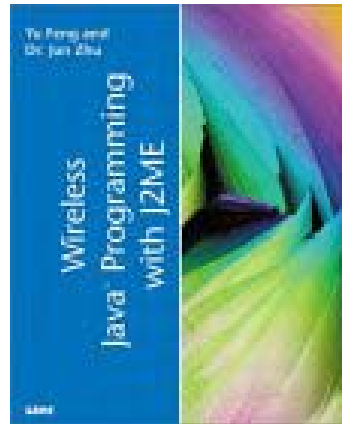
Mobile Information Device Profile for Java 2 MicroEdition

[http://www.amazon.com/exec/obidos/ASIN/0471034657/
ref=pd_sbs_b_3/107-1862323-6774146](http://www.amazon.com/exec/obidos/ASIN/0471034657/ref=pd_sbs_b_3/107-1862323-6774146)

Professional Java Mobile Programming

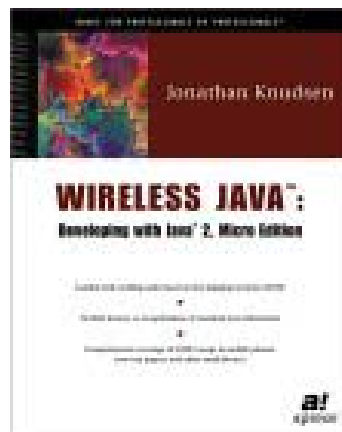
[http://www.amazon.com/exec/obidos/ASIN/1861003897/
qid=981191899/sr=1-8/ref=sc_b_8/105-2538596-2876714](http://www.amazon.com/exec/obidos/ASIN/1861003897/qid=981191899/sr=1-8/ref=sc_b_8/105-2538596-2876714)

Wireless Java Programming with J2ME



[http://www.amazon.com/exec/obidos/ASIN/0672321351/
ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R](http://www.amazon.com/exec/obidos/ASIN/0672321351/ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R)

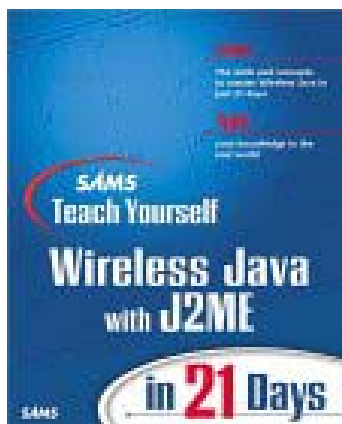
wireless Java : Developing with Java 2, Micro Edition



[http://www.amazon.com/exec/obidos/ASIN/189311550X/
qid=993487636/sr=1-2/ref=sc_b_2/107-1862323-6774146](http://www.amazon.com/exec/obidos/ASIN/189311550X/qid=993487636/sr=1-2/ref=sc_b_2/107-1862323-6774146)

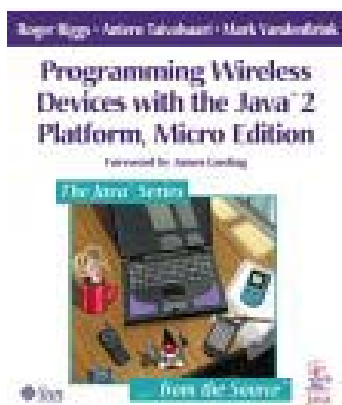
wireless Java with J2ME in 21 Days





http://www.amazon.com/exec/obidos/ASIN/0672321424/qid=993487433/sr=1-3/ref=sc_b_3/107-1862323-6774146

Programming Wireless Devices with the Java 2 Platform, Micro Edition



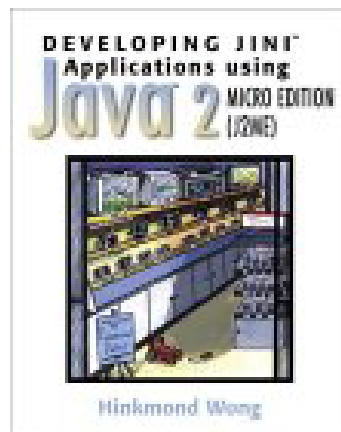
http://www.amazon.com/exec/obidos/ASIN/0201746271/ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R

Java 2 Micro Edition (J2ME) Application Development



[http://www.amazon.com/exec/obidos/ASIN/0672320959/
ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R](http://www.amazon.com/exec/obidos/ASIN/0672320959/ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R)

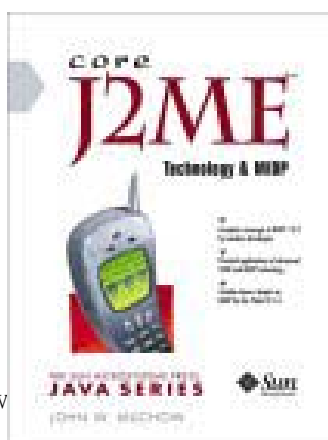
Developing Jini(tm) Applications Using Java(tm) 2 Micro Edition (J2ME)



[http://www.amazon.com/exec/obidos/ASIN/0201702444/
ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R](http://www.amazon.com/exec/obidos/ASIN/0201702444/ref=cm_mp_wl/002-3586460-8574424?colid=2MTDJHT8SC67R)



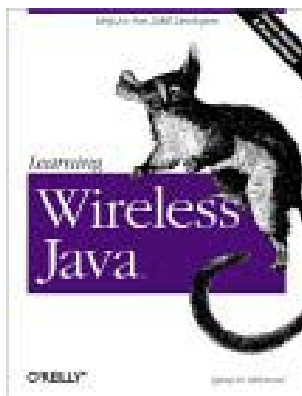
Core J2ME Technology

<http://www>

IN/0130669113/

ref=ase_billday/107-1964875-1306160

Core J2ME Technology



1-48



<http://www.amazon.com/exec/obidos/ASIN/0596002432/>

ref=ase_billday/107-1964875-1306160

Java 2 Micro Edition Application Development



<http://www.amazon.com/exec/obidos/ASIN/0072320959/>

ref=ase_billday/107-1964875-1306160

Instant Wireless Java with J2ME

<http://www.amazon.com/exec/obidos/ASIN/0072191759/>

ref=ase_billday/107-1964875-1306160

Wireless Programming with J2ME: Cracking the Code

<http://www.amazon.com/exec/obidos/ASIN/0764548859/>

ref=ase_billday/107-1964875-1306160



1

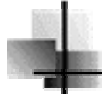
Java

Java 手機程式設計入門

Java



memo



A series of horizontal lines for writing, spanning the width of the page. The lines are evenly spaced and extend across the page, providing a structured area for notes.

