



# 使用者指南

---

Sun Java™ Wireless Toolkit for CLDC

2.5.2 版

Sun Microsystems, Inc.  
[www.sun.com](http://www.sun.com)

2007 年 9 月

Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

Sun Microsystems, Inc. 對本文件所述產品所採用的技術擁有相關智慧財產權。特別是 ( 但不僅限於 )，這些智慧財產權可能包含一項或多項在 <http://www.sun.com/patents> 上列出的美國專利，以及一項或多項美國及其他國家 / 地區的其他專利或申請中專利。

美國政府權利 — 商業軟體。政府使用者均應遵守 Sun Microsystems, Inc. 的標準授權合約和 FAR 及其增補文件中的適用條款。使用應遵守授權合約的條款。本發行物可能包含由協力廠商開發的材料。

Sun、Sun Microsystems、Sun 標誌、Java、Javadoc、Java Community Process、JCP、JDK、JRE、J2ME 與 J2SE 是 Sun Microsystems, Inc. 在美國及其他國家 / 地區的商標或註冊商標。

OpenGL 是 Silicon Graphics, Inc. 的註冊商標。

本服務手冊所涵蓋的產品和包含的資訊受到美國出口控制法規的控制，並可能受到其他國家 / 地區進出口法規的管轄。嚴禁將本產品直接或間接地用於核武器、飛彈、生化武器或海上核動力裝備，也不得將本產品直接或間接地提供給核武器、飛彈、生化武器或海上核動力裝備的一般使用者。嚴禁將本產品出口或再出口至美國禁運的國家 / 地區或美國出口排除清單中包含的實體 ( 包括但不僅限於被拒的個人和特別指定的國家 / 地區的公民清單 )。

本文件以其「原狀」提供，對任何明示或暗示的條件、陳述或擔保，包括對適銷性、特殊用途的適用性或非侵權性的暗示保證，均不承擔任何責任，除非此免責聲明的適用範圍在法律上無效。

# 目錄

---

前言 xiii

## 1. 簡介 1-1

- 1.1 關於多重使用者環境 1-1
  - 1.1.1 安裝目錄與工作目錄 1-1
  - 1.1.2 工作目錄檔案 1-2
- 1.2 快速入門 1-4
- 1.3 工具組元件 1-5
- 1.4 工具組功能 1-6
- 1.5 工具組更新 1-6
- 1.6 支援的技術 1-7

## 2. 開發 MIDlet 套裝軟體 2-1

- 2.1 關於專案 2-1
- 2.2 簡單的開發周期 2-3
  - 2.2.1 編輯原始碼 2-4
  - 2.2.2 生成 2-5
  - 2.2.3 執行 2-6

- 2.3 完整的開發周期 2-7
  - 2.3.1 封裝 2-8
  - 2.3.2 安裝 2-8
  - 2.3.3 執行 2-13
- 2.4 從 MIDlet 套裝軟體建立專案 2-13
- 2.5 使用混淆器 2-14
  - 2.5.1 安裝 ProGuard 2-14
  - 2.5.2 使用 ProGuard 2-14
- 2.6 使用除錯程式 2-15
- 2.7 在 Web 伺服器上部署應用程式 2-15
- 3. 使用專案 3-1
  - 3.1 選取 API 3-1
  - 3.2 變更 MIDlet 套件屬性 3-3
  - 3.3 處理 MIDlet 3-5
  - 3.4 使用推送註冊表 3-6
  - 3.5 設定內容處理器 3-7
  - 3.6 專案目錄結構 3-11
  - 3.7 使用第三方程式庫 3-11
    - 3.7.1 使用外部 API 3-12
    - 3.7.2 一個專案的第三方程式庫 3-13
    - 3.7.3 所有專案的第三方程式庫 3-13
  - 3.8 配置 Wireless Toolkit 3-13
    - 3.8.1 變更主控台字型 3-14
    - 3.8.2 設定應用程式目錄 3-14
    - 3.8.3 設定 javac 編碼特性 3-14
    - 3.8.4 使用修正控制系統 3-14

4.	使用模擬器	4-1
4.1	模擬器面板	4-1
4.2	模擬器控制	4-2
4.3	設定模擬器偏好	4-3
4.3.1	網路代理伺服器	4-3
4.3.2	儲存容量大小	4-4
4.3.2.1	永久性儲存裝置	4-4
4.3.2.2	堆疊大小	4-5
4.3.3	調整模擬器效能	4-5
4.4	暫停與繼續	4-7
4.5	單獨執行模擬器	4-7
4.6	使用第三方模擬器	4-8
5.	監視應用程式	5-1
5.1	使用效能評測器	5-1
5.1.1	調用圖	5-3
5.1.2	執行時間與呼叫次數	5-4
5.1.3	儲存與載入效能評測器資訊	5-4
5.2	使用記憶體監視器	5-4
5.2.1	儲存與載入記憶體監視器資訊	5-6
5.3	使用網路監視器	5-7
5.3.1	篩選訊息	5-8
5.3.2	排序訊息	5-8
5.3.3	儲存與載入網路監視器資訊	5-9
5.3.4	清除訊息樹狀結構	5-9

- 6. 安全性與 MIDlet 簽署 6-1
  - 6.1 權限 6-1
  - 6.2 選取安全性策略 6-3
    - 6.2.1 MSA 保護網域 6-3
    - 6.2.2 Java for the Wireless Toolkit Industry 保護網域 6-4
  - 6.3 簽署 MIDlet 套件 6-4
  - 6.4 管理金鑰 6-5
    - 6.4.1 建立新的金鑰組 6-5
    - 6.4.2 獲取真實金鑰 6-6
    - 6.4.3 匯入現有的金鑰組 6-7
    - 6.4.4 移除金鑰組 6-7
  - 6.5 管理憑證 6-7
    - 6.5.1 啟用和停用憑證 6-8
    - 6.5.2 匯入憑證 6-8
    - 6.5.3 移除憑證 6-9
  - 6.6 支援 USB 加密碟 6-9
    - 6.6.1 安裝 USB 加密碟驅動程式 6-9
    - 6.6.2 使用 USB 加密碟 6-10
- 7. 使用 Wireless Messaging API 7-1
  - 7.1 設定模擬器的電話號碼 7-1
  - 7.2 模擬不可靠的網路 7-3
  - 7.3 使用 WMA 主控台傳送訊息 7-3
    - 7.3.1 傳送文字的 SMS 訊息 7-4
    - 7.3.2 傳送二進位 SMS 訊息 7-5
    - 7.3.3 傳送文字或二進位 CBS 訊息 7-6
    - 7.3.4 傳送 MMS 訊息 7-6
  - 7.4 在 WMA 主控台中接收訊息 7-8
  - 7.5 與 WMA 一起使用網路監視器 7-8

<b>8. 使用 Mobile Media API</b>	<b>8-1</b>
8.1 支援的格式與協定	8-1
8.2 Adaptive Multi-Rate (AMR) 內容	8-2
8.2.1 Windows	8-2
8.2.2 Linux	8-2
8.2.2.1 啓用 AMR 支援	8-2
8.2.2.2 AMR 格式支援	8-3
8.3 使用 MediaControlSkin	8-3
8.4 媒體擷取	8-4
8.5 正常 MIDlet	8-4
8.6 鈴聲	8-4
8.6.1 下載鈴聲	8-4
8.6.2 鈴聲格式	8-5
<b>9. 使用 Mobile Graphics</b>	<b>9-1</b>
9.1 使用 Mobile 3D Graphics API	9-1
9.1.1 立即模式	9-2
9.1.2 保留模式	9-2
9.1.3 品質與速度之間的取捨	9-2
9.1.4 建立 Mobile 3D Graphics 內容	9-3
9.2 描繪可縮放式向量圖形內容	9-3
9.3 OpenGL® ES 簡介	9-4
<b>10. 使用 PIM 與 FileConnection API</b>	<b>10-1</b>
10.1 FileConnection API	10-1
10.2 PIM API	10-2
<b>11. 使用藍芽與 OBEX API</b>	<b>11-1</b>
11.1 藍芽模擬環境	11-1
11.2 透過紅外線使用 OBEX	11-1

- 11.3 設定 OBEX 與藍芽偏好 11-2
  - 11.3.1 OBEX 偏好 11-3
  - 11.3.2 藍芽內部特性 11-3
  - 11.3.3 藍芽系統特性 11-3
  - 11.3.4 藍芽 BCC 特性 11-4
- 12. 使用 Web 服務 12-1
- 13. 使用位置 API 13-1
  - 13.1 在執行階段設定模擬器的位置 13-1
  - 13.2 配置位置提供者 13-4
  - 13.3 設定地標 13-5
- 14. 使用 SATSA 14-1
  - 14.1 模擬器中的卡片槽 14-2
  - 14.2 使用 Java Card 平台模擬器 14-3
  - 14.3 使用具有 SATSA 的網路監視器 14-3
  - 14.4 調整存取控制 14-4
    - 14.4.1 指定 PIN 特性 14-5
    - 14.4.2 指定應用程式權限 14-5
    - 14.4.3 存取控制檔範例 14-7
- 15. 使用 SIP 15-1
  - 15.1 瞭解代理登錄和代理伺服器 15-1
  - 15.2 SIP 設定 15-2
  - 15.3 網路監視器中的 SIP 流量 15-3
  - 15.4 SIP 代理伺服器和代理登錄 15-4



<b>16. 使用 Payment API</b>	<b>16-1</b>
16.1 支付款項的專案設定	16-1
16.2 直接編輯支付款項屬性	16-4
16.3 支付款項偏好	16-4
16.4 檢視作業事件歷程	16-5
16.5 監視支付款項	16-7
<b>17. 使用 Mobile Internationalization API</b>	<b>17-1</b>
17.1 設定模擬器的語言環境	17-1
17.2 檢視應用程式資源	17-1
17.3 使用語言環境	17-3
17.4 使用資源檔案	17-3
17.5 使用資源	17-3
<b>A. 應用程式示範</b>	<b>A-1</b>
A.1 簡介	A-1
A.2 一般指示	A-3
A.3 Advanced Multimedia Supplements	A-4
A.4 藍芽示範	A-5
A.5 CHAPIDemo	A-7
A.6 CityGuide	A-10
A.7 示範	A-13
A.7.1 色彩	A-13
A.7.2 特性	A-13
A.7.3 Http	A-14
A.7.4 FontTestlet	A-14

A.7.5	股票	A-15
A.7.5.1	使用設定	A-15
A.7.5.2	股票追蹤器	A-15
A.7.5.3	What If?	A-16
A.7.5.4	警報	A-16
A.7.6	購票	A-16
A.7.7	ManyBalls	A-17
A.8	Demo3D	A-17
A.8.1	Life3D	A-17
A.8.2	PogoRoo	A-19
A.8.3	retainedmode	A-19
A.9	GoSIP	A-20
A.10	i18nDemo	A-22
A.11	JBricks	A-25
A.12	JSR172Demo	A-28
A.13	MobileMediaAPI	A-28
A.13.1	Simple Tones	A-29
A.13.2	Simple Player	A-29
A.13.3	Video	A-31
A.13.4	暫停音效測試	A-31
A.13.5	MobileMediaAPI 的屬性	A-32
A.14	網路示範	A-32
A.14.1	Socket Demo	A-32
A.14.2	Datagram Demo	A-34
A.15	ObexDemo	A-35
A.16	PDAPDemo	A-36
A.16.1	瀏覽檔案	A-36
A.16.2	PIM API	A-38

A.17	SATSADemos	A-40
A.17.1	APDUMIDlet	A-41
A.17.2	SATMIDlet	A-41
A.17.3	CryptoMIDlet	A-41
A.17.4	MohairMIDlet	A-42
A.18	SATSAJCRMIDemo	A-42
A.19	SIPDemo	A-43
A.20	SVGContactList	A-43
A.21	SVGDemo	A-45
A.21.1	SVG 瀏覽器	A-45
A.21.2	描繪 SVG 影像	A-45
A.21.3	播放 SVG 動畫	A-45
A.21.4	從頭開始建立 SVG 影像	A-46
A.21.5	跳跳球	A-46
A.21.6	最佳化的功能表	A-46
A.21.7	照片裝飾程式	A-47
A.21.8	定位服務	A-49
A.22	WMADemo	A-49
<b>B.</b>	<b>指令行參考</b>	<b>B-1</b>
B.1	先決條件	B-1
B.2	開發周期	B-1
B.2.1	生成	B-2
B.2.2	封裝	B-3
B.2.3	執行	B-4
B.2.4	除錯	B-5
B.3	啓動工具組 GUI 元件	B-6
B.4	設定模擬器偏好	B-6

B.5	使用安全功能	B-8
B.5.1	變更模擬器的預設保護網域	B-9
B.5.2	簽署 MIDlet 套裝軟體	B-9
B.5.3	管理憑證	B-10
B.6	使用 Stub 產生器	B-11
B.6.1	選項	B-11
C.	本土化	C-1
C.1	語言環境設定	C-1
C.2	模擬語言環境	C-2
C.3	字元編碼	C-2
C.4	Java 技術編譯器編碼設定	C-2
C.5	預設模擬器的字型支援	C-3
索引	索引	-1

# 前言

---

本文件描述了如何使用 Sun Java™ Wireless Toolkit for CLDC。

---

## 本書適用對象

本指南的目標使用者是使用 Sun Java™ Wireless Toolkit for CLDC 建立 Mobile Information Device Profile (MIDP) 應用程式的開發人員。本書不是 MIDP 程式設計的教學課程，也不是工具組所支援任何額外 API 的程式設計教學課程。您應該已經瞭解如何使用 Mobile Information Device Profile (MIDP) 與 Connected Limited Device Configuration (CLDC)。

如需 Java 程式設計語言入門方面的說明，請嘗試「New to Java Center」：

<http://java.sun.com/learning/new2java/>

若要 MIDP 程式設計的快速入門，請閱讀「Learning Path: Getting Started with MIDP 2.0」：

<http://developers.sun.com/techttopics/mobility/learn/midp/midp20/>

---

## 相關文件

本小節列出了相關的 Java Platform, Micro Edition (Java ME) 規格。您在某些規格名稱中可看到 Java ME 先前稱為 Java 2 Platform, Micro Edition (或 J2ME™)。雖然規格是重要的資訊，但有時不易取得。如需各種針對開發人員撰寫的文章，請嘗試 Sun 的 Mobility 網站：

<http://developers.sun.com/techttopics/mobility/>

表格 P-1 相關文件

主題	標題
自訂 Sun Java™ Wireless Toolkit for CLDC	Sun Java™ Wireless Toolkit for CLDC 基本自訂指南
版本說明	Sun Java™ Wireless Toolkit for CLDC 版本說明
CLDC 1.0 - JSR 30	J2ME Connected Limited Device Configuration
MIDP 1.0 - JSR 37	Mobile Information Device Profile for the J2ME Platform
PDAP Optional Packages - JSR 75	PDA Optional Packages for the J2ME Platform
藍芽與 OBEX - JSR 82	Java APIs for Bluetooth
MIDP 2.1 - JSR 118	Mobile Information Device Profile 2.0 (最終發行版本 2 稱為 MIDP 2.1)
CLDC 1.1 - JSR 139	J2ME Connected Limited Device Configuration
MMAPI - JSR 135	Mobile Media API
J2ME Web Services - JSR 172	J2ME Web Services Specification
SATSA - JSR 177	Security and Trust Services APIs for J2ME
Location API - JSR 179	Location API for J2ME
SIP API - JSR 180	SIP API for J2ME
Mobile 3D Graphics - JSR 184	Mobile 3D Graphics API for J2ME
JTWI - JSR 185	Java Technology for the Wireless Industry
WMA 2.0 - JSR 205	Wireless Messaging API (WMA)
CHAPI 1.0 - JSR 211	Content Handler API
SVG API - JSR 226	Scalable 2D Vector Graphics API for J2ME
Payment API - JSR 229	Payment API
Advanced Multimedia - JSR 234	Advanced Multimedia Supplements
Mobile Internationalization - JSR 238	Mobile Internationalization API
Java Binding for OpenGL® ES API - JSR 239	Java Binding for OpenGL® ES API
Mobile Service Architecture- JSR 248	Mobile Service Architecture

---

# 本書架構

本指南包括以下各章及附錄：

[第 1 章](#)介紹 Sun Java™ Wireless Toolkit for CLDC 及其提供的開發功能。

[第 2 章](#)描述建立與執行 MIDlet 的開發過程。

[第 3 章](#)說明如何使用工具組中的專案。您將學習如何調整專案特性、操作 MIDlet、使用推送註冊表以及瞭解專案目錄結構。

[第 4 章](#)描述模擬器，並說明如何調整其選項，及利用其許多功能。

[第 5 章](#)示範如何使用效能評測器、記憶體監視器與網路監視器，檢查應用程式的效能。

[第 6 章](#)描述如何簽署 MIDlet 套件，及管理金鑰與憑證。

[第 7 章](#)詳述執行與測試無線訊息傳送應用程式的支援。

[第 8 章](#)說明 Sun Java™ Wireless Toolkit for CLDC 如何支援 Mobile Media API。

[第 9 章](#)包含開發 3D 圖形內容的相關資訊。

[第 10 章](#)描述工具組如何實作存取本機檔案與個人資訊，例如連絡人與行事曆約會。

[第 11 章](#)涵蓋工具組的藍芽與 OBEX 模擬環境。

[第 12 章](#)示範如何使用 Web 服務 Stub 產生器。

[第 13 章](#)描述了如何使用模擬器的位置功能。

[第 14 章](#)將討論工具組對 SATSA 的支援。

[第 15 章](#)涵蓋工具組的 SIP 支援。

[第 16 章](#)描述了工具組的 Payment API 功能。

[第 17 章](#)說明了如何管理用於 Mobile Internationalization API 的資源。

[附錄 A](#) 描述 Sun Java™ Wireless Toolkit for CLDC 隨附的應用程式示範。

[附錄 B](#) 說明如何從指令行使用 Sun Java™ Wireless Toolkit for CLDC 的功能。

[附錄 C](#) 描述 Sun Java™ Wireless Toolkit for CLDC 中的本土化功能。

# 印刷排版慣例

表格 P-2 描述本文件中的字型使用方式。

表格 P-2 印刷排版慣例

字體	意義	範例
AaBbCc123 ( 固定間距 )	API 和語言元素、HTML 標記、網站 URL、指令名稱、檔案名稱、目錄路徑名稱、螢幕畫面輸出、範例代碼。	編輯您的 <code>.login</code> 檔案。 使用 <code>ls -a</code> 列出所有檔案。 <code>% You have mail.</code>
<b>AaBbCc123</b> ( 固定間距粗體 )	您鍵入的內容，與螢幕畫面輸出相對。	<code>% su</code> Password:
<i>AaBbCc123</i> ( 斜體 )	指令或路徑名稱中要由真實名稱或值替代的預留位置。	該檔案位於 <i>install-dir/bin</i> 目錄中。 這些稱為類別選項。
<b>AaBbCc123</b>	新的字彙或術語、要強調的詞。	請勿儲存此檔案。
「AaBbCc123」	用於書名及章節名稱。	請閱讀「使用者指南」中的第 6 章。
{AaBbCc.dir}	可變的檔案名稱與目錄。	在書中， <i>toolkit</i> 固定指 Sun Java™ Wireless Toolkit for CLDC 的安裝目錄。

# 存取 Sun 線上文件

下列網站提供有關 Java 技術的技術性文件：

- <http://developer.sun.com/>
- <http://java.sun.com/javame/>

# 我們歡迎您提出寶貴意見

我們致力於提高文件品質。您可以透過 [developers.sun.com](mailto:developers.sun.com) 電子郵件提供意見回饋。



# 簡介

---

本書介紹如何使用 Sun Java™ Wireless Toolkit for CLDC。

Sun Java™ Wireless Toolkit for CLDC 這套工具可供研發人員為行動電話與其他無線裝置開發應用程式。雖然 Sun Java™ Wireless Toolkit for CLDC 以 Mobile Information Device Profile (MIDP) 2.1 為基礎，但是它還支援許多選擇性套裝軟體，因此是一個功能頗為廣泛的開發工具組。

---

## 1.1 關於多重使用者環境

Sun Java Wireless Toolkit 2.5.2 for CLDC 可安裝在執行 Windows 或 Linux 的受支援版本的系統上。所有在主機上具有帳號的使用者均可存取該工具組，既可單獨存取，也可多個同時存取。

### 1.1.1 安裝目錄與工作目錄

Windows 路徑中包含磁碟機字母，並使用反斜線作為目錄分隔符號。Linux 路徑使用正斜線。在 Linux 路徑中，~ 表示 Linux 使用者的主目錄。

為支援多個使用者，工具組會建立一個安裝目錄，當作複製的來源使用。本文件使用變數 *workdir* 來表示工具組工作目錄，並使用 *toolkit* 來表示安裝目錄。各使用者的個人檔案儲存於名為 *j2mewtk* 的獨立工作目錄中，對於安裝的各版本，該目錄下均包含一個相應的子目錄。*workdir* 的預設位置通常為以下位置之一：

---

**Windows:** C:\Documents and Settings\User\j2mewtk\2.5.2  
(其中 *User* 為您的帳號名稱)

**Linux:** ~/j2mewtk/2.5.2 (其中 ~ 為您的主目錄)

---

文件中說明了 Windows 與 Linux 之間的顯著差異。如果僅提及一個作業系統，則您可以假定上述路徑差異是適用的。

## 1.1.2 工作目錄檔案

安裝時，安裝程式會將這些檔案的子集從安裝目錄複製到每個使用者的工作目錄。工作目錄內容如下：

- **.settings** — 最初包含 `security.properties`。
- **appdb** — 將整個應用程式資料庫複製到您的工作目錄。
- **apps** — `apps` 目錄在您的工作目錄建立時為空白。如果您開啓一個示範專案，則會將其自動複製到該位置。當您建立新專案時（第 2-1 頁的第 2.1 節「關於專案」），會將該專案儲存於此。
- **wtllib** — 包含模擬器特性檔（請參閱第 4 章）以及 HTTP 與 WMA 伺服器與用戶端的狀態資訊。
- **sessions** — 此為監視工具的預設儲存目標（請參閱第 5 章）。最初為空白。

示範應用程式的所有原始碼均可在 `toolkit\apps` 中找到，每個示範均有其自己的專案目錄。所有使用者都可看到放置於 `toolkit\apps` 中的所有應用程式。

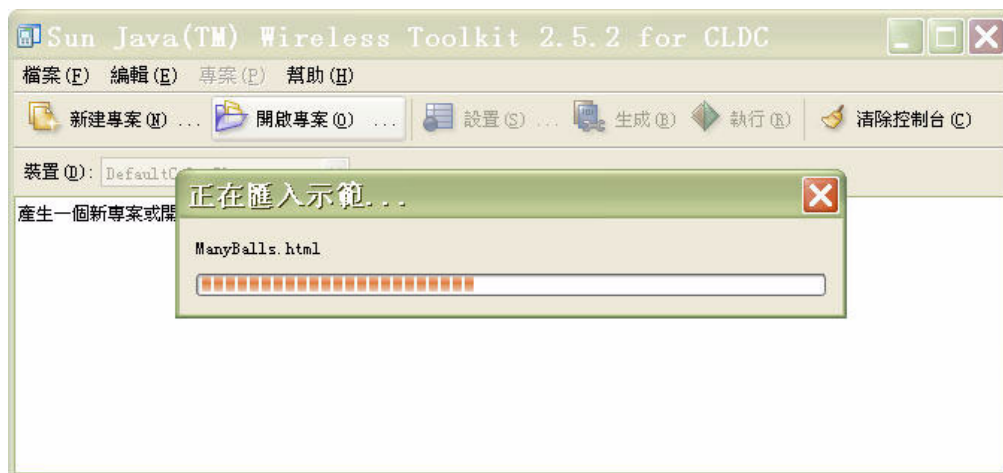
例如在 Windows 中，`MobileMediaAPI` 示範的原始碼存放在 `toolkit\apps\MobileMediaAPI\src` 中。使用 [開啓專案] 指令時，您可以看到安裝目錄與工作目錄中的專案（如果存在）。如圖 1-1 中所示，本機目錄中的專案以粗體字型顯示，安裝目錄中的專案以斜體字型顯示。若要僅顯示本機檔案，請清除 [顯示可用示範] 核取方塊。

圖 1-1 專案清單以粗體顯示本機專案



當您從安裝目錄開啟專案時，會將應用程式副本匯入至 `workdir\apps`。通常，您察覺不到匯入程序，但是在較慢的電腦上您可能看到如圖 1-2 所示的進度對話方塊。任何自訂都應針對您的本機副本進行。

圖 1-2 從 toolkit 匯入專案至 workdir



## 1.2 快速入門

若要立即開始，可嘗試 Sun Java™ Wireless Toolkit for CLDC 隨附的示範應用程式。

若要執行示範，請按以下程序啓動此工具組。

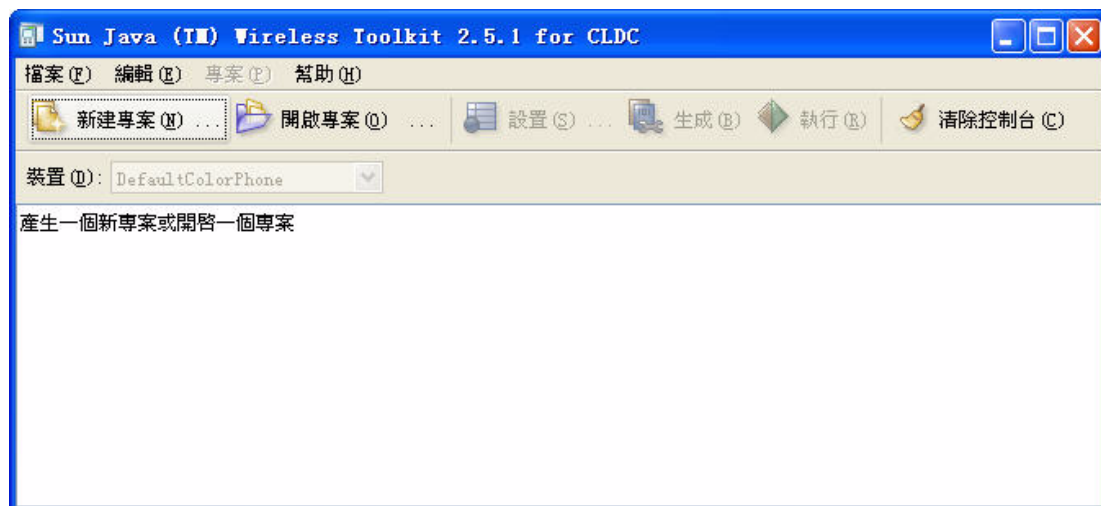
- 在 Microsoft Windows 中，選擇 [ 開始 ] > [ 程式集 ] > [ Sun Java Wireless Toolkit 2.5.2 for CLDC ] > [ Wireless Toolkit 2.5.2 ]。<sup>1</sup>
- 在 Linux 中，將目錄變更為 `toolkit/bin`。若要執行此工具組，請輸入 `./ktoolbar`。

會開啓一個類似圖 1-3 的視窗。

---

1. 依 Microsoft Windows 配置方式的不同，您可能不是選擇 [ 開始 ] > [ 程式集 ]，而要選擇 [ 開始 ] > [ 所有程式 ]。

圖 1-3 工具組使用者介面



按一下【開啟專案】按鈕以查看所有可用應用程式的清單。如圖 1-1 中所示，斜體名稱表示儲存在安裝目錄中的應用程式；粗體名稱（如果有）表示儲存在您的工作目錄中的應用程式。從中挑選一個專案，並按一下對話方塊中的【開啟專案】按鈕。如果您之前未開啟過此專案，則將會在您的工作目錄中建立副本，如第 1-2 頁的第 1.1.2 節「[工作目錄檔案](#)」中所討論。

應用程式開啟後，請按【執行】按鈕。將顯示執行範例應用程式的模擬器。

大部分示範本身就具備足夠的說明，但部分示範另外有其他指示。某些示範要求您使用【專案】>【經由 OTA 執行】而非【執行】。如需示範的一般說明與描述，請參閱[附錄 A](#)。

## 1.3 工具組元件

Sun Java™ Wireless Toolkit for CLDC 有三個主要元件：

- 使用者介面自動執行建立 MIDP 應用程式的多項相關工作。
- 模擬器是一個模擬的行動電話。在測試 MIDP 應用程式時非常有用。
- 公用程式集提供了其他有用的功能，包括文字傳訊主控台和加密公用程式。

您可以從使用者介面生成應用程式、啟動模擬器及啟動公用程式。或者，您可以讓模擬器和公用程式自行執行，這在許多狀況下非常有用。比如說，如果要示範 MIDP 應用程式，就可以單獨執行模擬器。

您只需要一個額外的工具，就是文字編輯器，以編輯原始碼。

---

## 1.4 工具組功能

Sun Java™ Wireless Toolkit for CLDC 支援建立具有下列主要功能的 MIDP 應用程式：

- **生成與封裝：**您只須撰寫原始碼，其他工作都可以交由工具組負責處理。只要按一下按鈕，工具組就會編譯原始碼，預先驗證類別檔案，並封裝成 MIDlet 套件。
- **執行與監視：**您可以直接在模擬器中執行 MIDlet 套件，也可以使用類似於在實際裝置上安裝應用程式的方式安裝它。本工具組提供記憶體監視器、網路監視器與方法效能評測器，以分析 MIDlet 的作業。
- **MIDlet 套件簽署：**工具組包含一些工具，以加密簽署 MIDlet 套件。這可以在不同的保護網域中，測試 MIDlet 的作業。

---

## 1.5 工具組更新

安裝時，您可以選擇啟動 [ 檢查產品更新 ] 功能。此功能使用網路每隔七天檢查一次更新。為了確定是否需要更新，此功能將收集以下資訊：

- 作業系統
- 電腦中的國家 / 地區及語言設定
- 日期及目前 Wireless Toolkit 的版本
- 由 Wireless Toolkit 產生的唯一隨機使用者 ID

此資料用於改進產品。請閱讀 Sun 的隱私權政策 (<http://www.sun.com/privacy/index.html>)。

若要啟動或關閉此功能，請選取 [ 編輯 ] > [ 偏好 ]，然後按一下 [ 網路配置 ]。核取或取消核取面板底部的 [ 檢查更新 ] 方塊。

## 1.6 支援的技術

Sun Java™ Wireless Toolkit for CLDC 支援透過 Java Community Process™ (JCP™) 程式定義的許多標準應用程式設計介面 (API)。表格 1-1 顯示這些 API，並包含其規格的連結。

表格 1-1 支援的 JCP 程式 API

JSR API	名稱 URL
JSR 248 MSA 1.0	Mobile Service Architecture <a href="http://jcp.org/en/jsr/detail?id=248">http://jcp.org/en/jsr/detail?id=248</a>
JSR 185 JTWI 1.0	Java Technology for the Wireless Industry <a href="http://jcp.org/en/jsr/detail?id=185">http://jcp.org/en/jsr/detail?id=185</a>
JSR 139 CLDC 1.1	Connected Limited Device Configuration <a href="http://jcp.org/en/jsr/detail?id=139">http://jcp.org/en/jsr/detail?id=139</a>
JSR 118 MIDP 2.0	Mobile Information Device Profile <a href="http://jcp.org/en/jsr/detail?id=118">http://jcp.org/en/jsr/detail?id=118</a>
JSR 75 PIM 與檔案	PDA Optional Packages for the J2ME Platform <a href="http://jcp.org/en/jsr/detail?id=75">http://jcp.org/en/jsr/detail?id=75</a>
JSR 82 藍芽與 OBEX	Java APIs for Bluetooth <a href="http://jcp.org/en/jsr/detail?id=82">http://jcp.org/en/jsr/detail?id=82</a>
JSR 135 MMAPI 1.1	Mobile Media API <a href="http://jcp.org/en/jsr/detail?id=135">http://jcp.org/en/jsr/detail?id=135</a>
JSR 172	J2ME Web Services Specification <a href="http://jcp.org/en/jsr/detail?id=172">http://jcp.org/en/jsr/detail?id=172</a>
JSR 177 SATSA	Security and Trust Services API for Java ME <a href="http://jcp.org/en/jsr/detail?id=177">http://jcp.org/en/jsr/detail?id=177</a>
JSR 179 位置	Location API for Java ME <a href="http://jcp.org/en/jsr/detail?id=179">http://jcp.org/en/jsr/detail?id=179</a>
JSR 180 SIP	SIP API for Java ME <a href="http://jcp.org/en/jsr/detail?id=180">http://jcp.org/en/jsr/detail?id=180</a>
JSR 184 3D 圖形	Mobile 3D Graphics API for J2ME <a href="http://jcp.org/en/jsr/detail?id=184">http://jcp.org/en/jsr/detail?id=184</a>
JSR 205 WMA 2.0	Wireless Messaging API <a href="http://jcp.org/en/jsr/detail?id=205">http://jcp.org/en/jsr/detail?id=205</a>

表格 1-1 支援的 JCP 程式 API ( 繼續 )

JSR API	名稱 URL
JSR 211 CHAPI	Content Handler API <a href="http://jcp.org/en/jsr/detail?id=211">http://jcp.org/en/jsr/detail?id=211</a>
JSR 226	Scalable 2D Vector Graphics API for J2ME <a href="http://jcp.org/en/jsr/detail?id=226">http://jcp.org/en/jsr/detail?id=226</a>
JSR 229	Payment API <a href="http://jcp.org/en/jsr/detail?id=229">http://jcp.org/en/jsr/detail?id=229</a>
JSR 234 AMMS	Advanced Multimedia Supplements <a href="http://jcp.org/en/jsr/detail?id=234">http://jcp.org/en/jsr/detail?id=234</a>
JSR 238 MIA	Mobile Internationalization API <a href="http://jcp.org/en/jsr/detail?id=238">http://jcp.org/en/jsr/detail?id=238</a>
JSR 239	Java Binding for OpenGL® ES API <a href="http://jcp.org/en/jsr/detail?id=239">http://jcp.org/en/jsr/detail?id=239</a>
JSR 248	Mobile Service Architecture <a href="http://jcp.org/en/jsr/detail?id=248">http://jcp.org/en/jsr/detail?id=248</a>



## 開發 MIDlet 套裝軟體

---

這一章描述如何使用 Sun Java™ Wireless Toolkit for CLDC 建立應用程式。一開始，先說明工具組專案，然後逐步完成開發過程。

建立 MIDlet 套裝軟體應用程式時，您可能會遵循兩個基本的開發周期。第一個開發周期比較快，也比較簡單。在最初開發階段也非常有用。第二個周期比較長，但有比較寬裕的時間進行全方位而且實際的測試。

這一章後面的部分會介紹，如何將 Sun Java™ Wireless Toolkit for CLDC 與進階開發工具（如混淆器與除錯程式）結合使用。最後一節簡要說明如何配置 Web 伺服器，以服務 MIDP 應用程式。

---

### 2.1 關於專案

在 Sun Java™ Wireless Toolkit for CLDC 中，MIDlet 套裝軟體有條理的組織成一個個專案，而一個專案最後的成果就是一個 MIDlet 套裝軟體。專案包含即將用來生成 MIDlet 套裝軟體的所有檔案，包括 Java 原始碼檔案、資源檔案與 MIDlet 描述元。

Sun Java™ Wireless Toolkit for CLDC 一次處理一個專案。您可以建立新專案，或開啓現有專案。

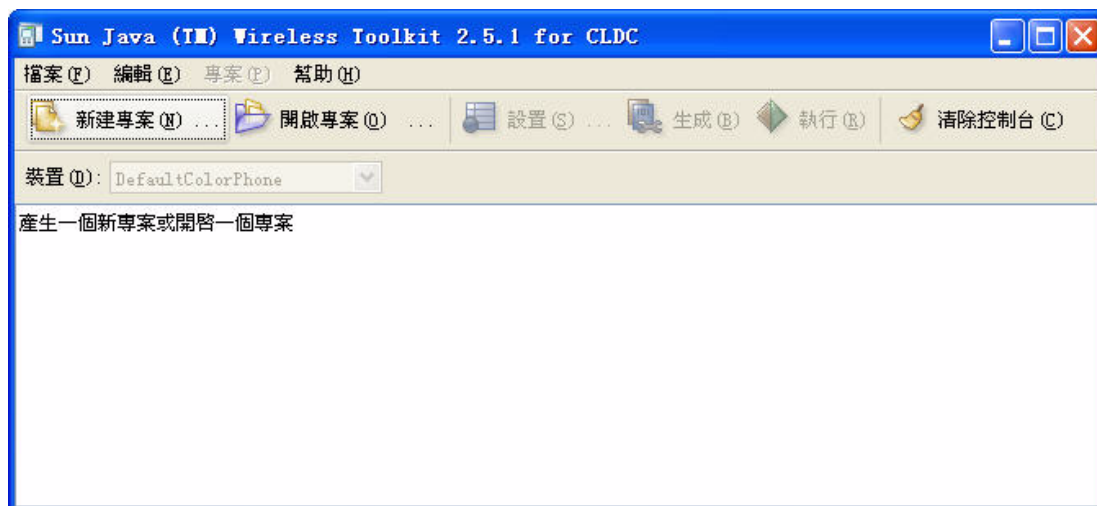
這一章會使用非常簡單的專案範例。當您研讀開發周期的每個步驟時，您可以使用工具組一起進行相同的操作。

若要建立新專案，請先啓動使用者介面。在 Microsoft Windows 中，選擇 [開始] > [程式集] > [Sun Java Wireless Toolkit 2.5.2 for CLDC] > [Wireless Toolkit 2.5.2]。<sup>1</sup> 將顯示使用者介面，如圖 2-1 中所示。

---

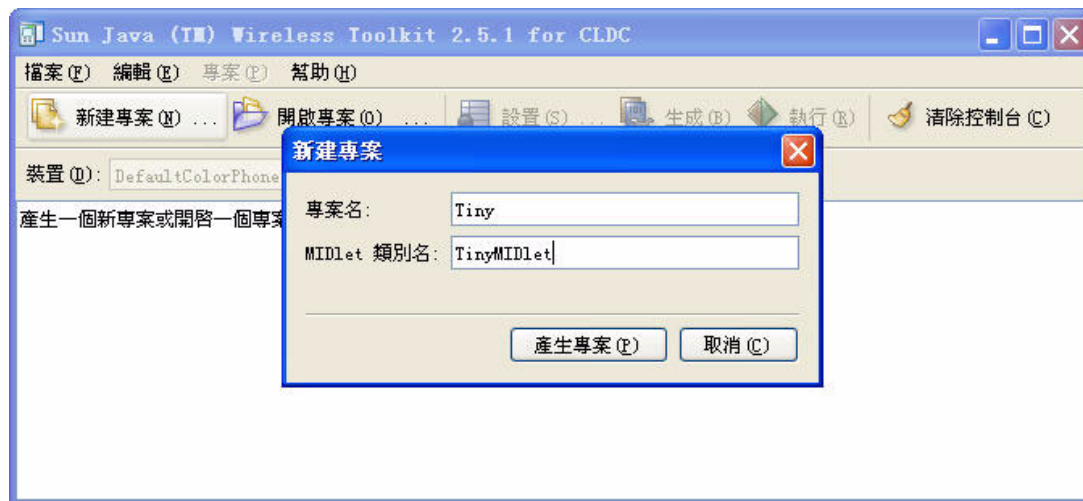
1. 依 Microsoft Windows 配置方式的不同，您可能不是選擇 [開始] > [程式集]，而要選擇 [開始] > [所有程式]。

圖 2-1 工具組的使用者介面



按一下 [ 新建專案 ]。工具組會要求您輸入專案名稱與即將撰寫的 MIDlet 類別名稱。填入名稱，再按一下 [ 建立專案 ]。

圖 2-2 建立新專案



顯示 [ 設定 ] 視窗。您的選擇會影響專案的生成環境。此範例使用預設選項即可，所以只要按一下 [ 確定 ] 關閉視窗。主控台上將顯示一些訊息，告訴您此專案的原始碼與資源檔案確切的儲存位置。

圖 2-3 主控台上的檔案位置



## 2.2 簡單的開發周期

簡單的開發周期如下：

編輯原始碼 > 生成 > 執行

### 1. 編輯原始碼。

在這個步驟中，要建立應用程式即將使用的 Java 原始檔與資源檔案。

### 2. 生成。

工具組會編譯並預先驗證您的 Java 原始檔。

### 3. 執行。

已編譯的 Java 類別檔案在模擬器上執行。

當工具組嘗試編譯原始碼檔案時，如果發生錯誤，請回頭重新編輯。如果在模擬器中測試應用程式時發現錯誤，請編輯該原始碼檔案以修正錯誤。

概略瞭解簡單的開發周期後，這一節其他部分會演示如何使用 Sun Java™ Wireless Toolkit for CLDC 完成每個步驟。

## 2.2.1 編輯原始碼

編輯原始碼是唯一完全無需使用 Sun Java™ Wireless Toolkit for CLDC 的步驟。使用自選的文字編輯器，建立與編輯原始碼檔案。如果您沒有喜歡的文字編輯器，請嘗試 jEdit (位於 <http://jedit.org/>)。

如果您正按照範例專案進行操作，請建立一個新的 Java 技術原始碼檔案 TinyMIDlet.java。將其儲存在專案的來源目錄中。例如，在 Windows 上此檔案為 *workdir\apps\Tiny\src\TinyMIDlet.java*。該檔案包含以下簡單的 MIDlet：

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class TinyMIDlet
    extends MIDlet
    implements CommandListener {
    public void startApp() {
        Display display = Display.getDisplay(this);

        Form mainForm = new Form("TinyMIDlet");
        mainForm.append("Welcome to the world of MIDlets!");

        Command exitCommand = new Command("Exit", Command.EXIT, 0);
        mainForm.addCommand(exitCommand);
        mainForm.setCommandListener(this);

        display.setCurrent(mainForm);
    }
    public void pauseApp () {}

    public void destroyApp(boolean unconditional) {}

    public void commandAction(Command c, Displayable s) {
        if (c.getCommandType() == Command.EXIT)
            notifyDestroyed();
    }
}
```

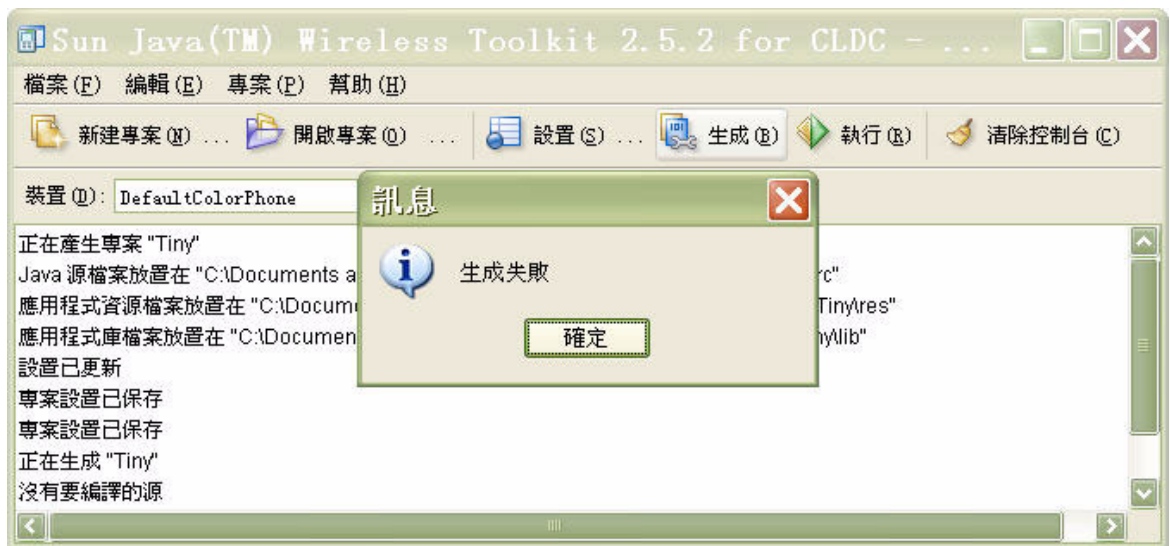
完成時，把檔案儲存起來。

## 2.2.2 生成

下一個步驟是生成原始碼。工具組讓這個部分非常容易。

在使用者介面中，按一下【生成】按鈕。假設您把原始碼檔案儲存在正確的位置，工具組將會找到檔案，並進行編譯。編譯錯誤會顯示在主控台中。如果有錯誤，如圖 2-4 中所示，請編輯原始碼以修正問題。排除錯誤之後，主控台會通知您生成程序已成功完成。

圖 2-4 有關生成的訊息

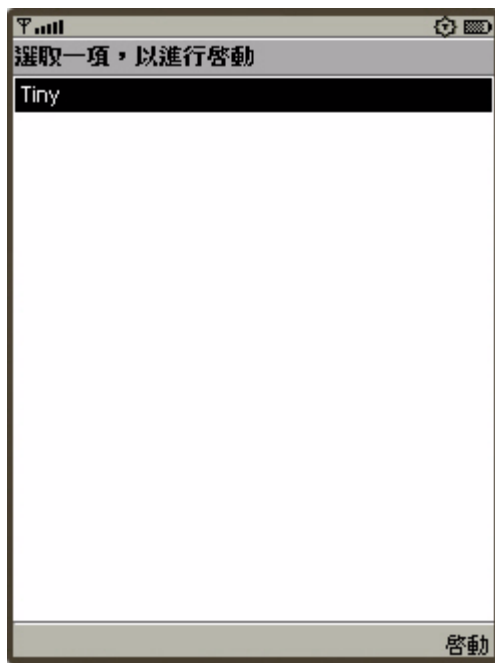


工具組也會在幕後預先驗證已編譯的類別檔案。MIDlet 類別檔案必須經過預先驗證後，才能在 MIDP 裝置或模擬器上執行。工具組會自動為您處理此細節。如需有關預驗證的詳細資訊，請參閱 CLDC 規格。

## 2.2.3 執行

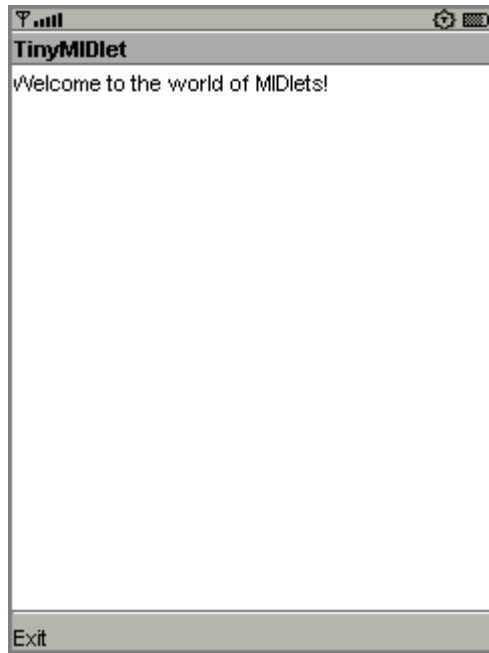
生成成功之後，專案就已經完成準備工作，可以在模擬器中試用。按一下【執行】按鈕。模擬器會顯示專案中所有 MIDlet 清單。

圖 2-5 專案 MIDlet 清單



選擇您要的 MIDlet，並選取【啟動】。如果您要照著 TinyMIDlet 範例來做，結果將如圖 2-6 中所示。

圖 2-6 TinyMIDlet 實際運作



## 2.3 完整的開發周期

第二個開發周期稍微複雜一點。它由下列主要步驟組成：

編輯原始碼 > 封裝 > 安裝 > 執行

1. 編輯原始碼。

這與簡單周期相同。

2. 封裝。

在此步驟中，Sun Java™ Wireless Toolkit for CLDC 會編譯並預先驗證原始碼檔案（基本上，跟之前的生成步驟一樣）。然後將 Java 類別檔案與資源檔案結合起來，成為 MIDlet 套裝軟體 Java Archive (JAR) 檔與 MIDlet 套裝軟體描述元。

3. 安裝。

MIDlet 套裝軟體需要安裝後才能執行。您可以將 MIDlet 套裝軟體安裝到 Sun Java™ Wireless Toolkit for CLDC 模擬器或實際裝置上。

#### 4. 執行。

與在簡單開發周期一樣，執行應用程式，並測試有無錯誤。

在完整開發周期中，第一個步驟與簡單開發周期完全一樣。編輯原始碼的方式都相同。生成步驟現在併入封裝中。

完整開發周期包含封裝與安裝二個新的步驟。最後，在許多重要的方面，安裝應用程式的執行方式不同於簡單開發周期中應用程式的執行方式。

## 2.3.1 封裝

Sun Java™ Wireless Toolkit for CLDC 可將 MIDlet 套裝軟體的封裝作業自動化。封裝出來的成果是二個檔案，一個 MIDlet 描述元與一個 MIDlet 套裝軟體 JAR 檔案。描述元是一個小文字檔，其中包含有關 MIDlet 套裝軟體的資訊。JAR 檔案包含組成 MIDlet 套裝軟體的類別檔案與資源。裝置在下載整個 JAR 檔案之前，可以使用描述元瞭解應用程式，這對於記憶體不足、頻寬又有限的無線世界而言，是很重要的考量。

若要讓工具組封裝您的 MIDlet 套裝軟體，請選擇 [ 專案 ] > [ 封裝 ] > [ 建立封裝 ]。就會產生 MIDlet 套裝軟體描述元與 JAR 檔案，並放在專案的 bin 目錄中。

封裝可能需要其他步驟。您可以使用程式碼混淆器，將 MIDlet 套裝軟體 JAR 檔案縮小，這個技術會在本章稍後說明。此外，工具組提供了一些工具可讓您加密簽署 MIDlet 套裝軟體。如需詳細資訊，請參閱第 6 章。

## 2.3.2 安裝

若要正確測試 MIDlet 套裝軟體，將它安裝在工具組模擬器或實際裝置上。當您在使用者介面中按下 [ 執行 ] 按鈕後，MIDlet 套裝軟體不會安裝到模擬器中。而是由模擬器直接執行 MIDlet 類別。

模擬器還能夠將應用程式安裝在其記憶體中，整個過程類似在實際裝置上透過 Over the Air (OTA) 傳送與安裝應用程式。若要在 Sun Java™ Wireless Toolkit for CLDC 模擬器中安裝應用程式，請選擇 [ 專案 ] > [ 經由 OTA 執行 ]。

模擬器視窗隨即開啓，但模擬器不會直接執行您的 MIDlet 類別，而是顯示 [ 應用程式管理軟體 (AMS) ] 歡迎畫面。模擬器的軟體是實際裝置在管理 MIDlet 套裝軟體時必備的一種軟體類型。



圖 2-7 模擬器 AMS 歡迎畫面



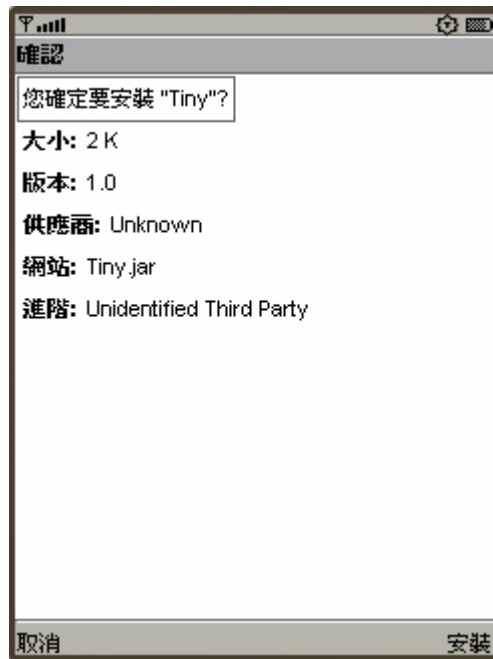
選擇【應用程式】進入已安裝應用程式的主要清單。選取【安裝應用程式】，再按下模擬器上的選取按鈕。模擬器會提示您輸入要安裝的應用程式 URL 位置。URL 已經預先完成。

圖 2-8 URL 提示



從功能表選擇 [ 開始 ]，開始安裝。模擬器會顯示它在 URL 找到的應用程式清單。選擇唯一的一個，再從功能表選取 [ 安裝 ]。模擬器會最後一次確認您的意向。

圖 2-9 確認安裝



再次選擇 [ 安裝 ] 以完成安裝。您會回到模擬器的已安裝應用程式清單，清單中會包含您剛才新增的應用程式。

圖 2-10 應用程式功能表



[經由 OTA 執行] 是極為有用的功能，讓您可以在工具組模擬器上輕鬆安裝 MIDlet 套件。有些功能一定要使用這個技術進行測試，包括推送註冊表以及已簽署 MIDlet 套裝軟體的安裝。

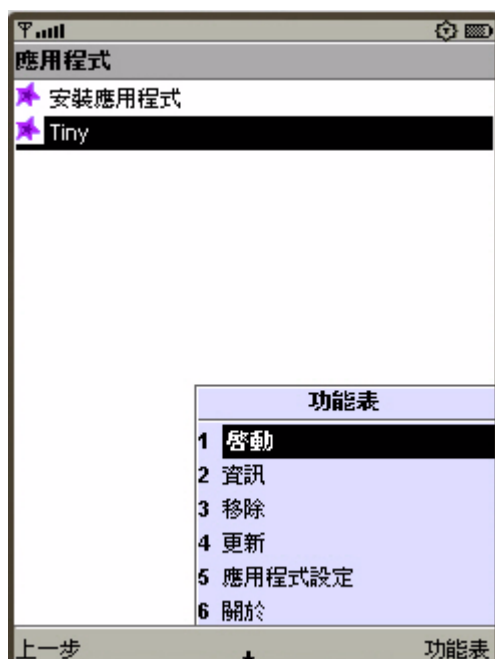
如果要在實際裝置上測試 MIDlet 套裝軟體，您必須先進行安裝。這項動作的進行方式，絕大部分須視所用裝置而定。可能的狀況有二：

- 您可以在 Web 伺服器上部署應用程式，然後使用 MIDP 2.0 規格中描述的 Over the Air (OTA) 通訊協定，將應用程式從伺服器傳送到裝置。當使用者購買或安裝您的應用程式時，最可能採用這種機制。
- 您可以利用藍芽、紅外線或串列連線，將 MIDlet 套裝軟體傳送到裝置。這種方式比執行 Web 伺服器簡單一點，而且雖然無法讓您觀察到經由 OTA 在裝置上安裝應用程式的整個過程，卻可讓您看到應用程式在裝置上的執行效果。

### 2.3.3 執行

安裝應用程式後，請從清單中選擇此應用程式，然後從功能表中選擇 [ 啟動 ]。

圖 2-11 啟動安裝的應用程式



在實際裝置上執行應用程式的方式，絕大部分須視裝置本身而定。如需詳細資訊，請參閱裝置的文件。

## 2.4 從 MIDlet 套裝軟體建立專案

您也可以使用 MIDlet 套裝軟體歸檔 (.jar 檔案) 和描述元 (.jad 檔案) 來建立 Sun Java™ Wireless Toolkit for CLDC 專案。即使您沒有可用的原始碼，這對於從工具組使用者介面執行 MIDlet 套裝軟體也是非常有用的。您可以使用工具組輕鬆處理描述元中的屬性，或者可以執行專案並使用工具組的監視工具 (如第 5 章所述) 檢查其運作方式。

若要以 MIDlet 套裝軟體為基礎建立專案，請選擇 [ 檔案 ] > [ 從 JAD/JAR 建立專案 ]。瀏覽至您要使用的描述元，按一下 [ 開啟 ]。請注意，描述元和 JAR 檔案必須在同一目錄下。

---

## 2.5 使用混淆器

**混淆器 (Obfuscator)** 是可以將類別檔案變小的工具。MIDlet 套裝軟體必須非常精簡，一來可以縮短下載時間，二來也可以符合部分製造商或供應商對 JAR 檔案大小所設下的嚴格限制。使用混淆器就是讓 MIDlet 套裝軟體 JAR 檔案變小的一種方法（但不是唯一的方法）。

您可以在開發周期的封裝步驟中使用混淆器。雖然 Sun Java™ Wireless Toolkit for CLDC 不隨附混淆器，但已經配置成可以使用 ProGuard 混淆器。您只需下載 ProGuard，並將它放在工具組找得到的位置即可。

ProGuard 的發行遵循 GNU 一般大眾授權 (General Public License, GPL) 的規定。如果您同意授權規定，即可免費下載及使用 ProGuard。

### 2.5.1 安裝 ProGuard

請遵循以下步驟安裝 ProGuard：

1. 移至 **ProGuard** 網站 <http://proguard.sourceforge.net/>。
2. 下載最新版本。
3. 從 **ProGuard** 安裝的 lib 目錄，將 `proguard.jar` 檔案解壓縮到 `toolkit/bin` 目錄。

### 2.5.2 使用 ProGuard

安裝 ProGuard 之後，只要選擇 [ 專案 ] > [ 封裝 ] > [ 建立模糊化封裝 ] 即可開始使用。

在某些狀況下，您必須提供控制混淆器運作方式的程序檔檔案。例如，如果是用 `Class.forName()` 載入類別，您必須告訴 ProGuard 不要變更類別名稱。

若要調用該程序檔，您必須能夠修改您自己的 `ktools.properties` 檔案副本。

- 將 `toolkit/wtklib/os/ktools.properties` 複製至 `workdir/wtklib/os/ktools.properties`。

請用文字編輯器建立程序檔檔案，然後儲存在專案的主目錄下。如需程序檔檔案的資訊，請參閱 ProGuard 的文件。接下來，告訴工具組如何找到此檔案。若要執行此動作，請編輯您已複製到 `workdir/wtklib` 中的 `ktools.properties` 檔案。新增底下這一行：

```
obfuscate.script.name: scriptfile
```

用您使用的程序檔檔案名稱取代 `scriptfile`。您必須退出並重新啟動工具組才能使變更生效。

---

## 2.6 使用除錯程式

應用程式除了自行執行之外，還能夠使用除錯程式來執行。除錯程式可讓您更密切監視執行中的應用程式、設定中斷點與檢視變數。

您必須自行提供除錯程式。您可以使用 Java SE 平台的 `jdb` 除錯程式，或您選擇的其他除錯程式。如果您想要使用除錯程式，則可納入 Sun Java™ Wireless Toolkit for CLDC 的整合開發環境 (IDE) 可能是最好的選擇 (如 Sun Java Studio Mobility 軟體)。如需更多資訊，請參閱

<http://www.sun.com/software/products/jsmobility/>。

由選擇 [ 專案 ] > [ 除錯 ] 開始。輸入除錯程式用來與模擬器連線的 TCP/IP 連接埠號。按一下 [ 除錯 ]。模擬器開始執行，並等待來自除錯程式的連接。

啟動除錯程式，將它連接到您指定的連接埠。務必將遠端除錯程式設成以遠端模式執行，而且使用 TCP/IP。如需詳細資訊，請參閱除錯程式的文件。

「除錯 MIDlet」具有在 Sun Java™ Wireless Toolkit for CLDC 中使用 `jdb` 的相關資訊。可以在

<http://developers.sun.com/techtopics/mobility/midp/questions/jdb/> 取得。

---

## 2.7 在 Web 伺服器上部署應用程式

MIDP 2.0 規格包括 Over The Air User Initiated Provisioning Specification，以描述 MIDlet 套裝軟體如何透過 Over-the-Air (OTA) 傳輸到裝置。您可以使用 Sun Java™ Wireless Toolkit for CLDC 模擬器測試這種狀況。

若要在 Web 伺服器上從遠端部署封裝的 MIDP 應用程式，請將 Java Application Descriptor (JAD) 檔的 MIDlet-Jar-URL 特性變更為 JAR 檔案的 URL。URL 必須是絕對路徑。例如：

MIDlet-Jar-URL: <http://your.server.com/midlets/example.jar>

接著，確定 Web 伺服器使用 JAD 和 JAR 檔案正確的 MIME 類型：

- 對於 MIDlet 套裝軟體描述元，將 `.jad` 副檔名對映到 `text/vnd.sun.j2me.app-descriptor` MIME 類型。
- 對於 MIDlet 套裝軟體 JAR 檔案，將 `.jar` 副檔名對映到 `application/java-archive` MIME 類型。

關於如何配置 Web 伺服器的詳細資訊，是依據所使用的特定軟體而定。

模擬器會在 OTA 供裝過程中實作裝置的運作方式。您可以使用模擬器，測試並示範 MIDlet 套裝軟體從伺服器到裝置的完整佈建程序。所需要的執行動作，只是啟動模擬器的 AMS 即可。如果用過 [ 經由 OTA 執行 ] 選項，則您可能已經熟悉了 AMS。

若要啟動模擬器的 AMS，您有二個選項：

- 在 Microsoft Windows 的 [ 開始 ] 功能表中，選擇 [ 開始 ] > [ 程式集 ] > [ Sun Java Wireless Toolkit 2.5.2 for CLDC ] > [ OTA 供裝 ]。
- 從指令行發出以下指令：

---

**Windows:**    `toolkit\bin\emulator -Xjam`

**Linux:**        `toolkit/bin/emulator -Xjam`

---

接著，依照 AMS 提示安裝您的應用程式。這個程序非常類似本章描述的 [ 經由 OTA 執行 ] 選項，但是您必須輸入自己的伺服器 URL，以安裝您的應用程式。



## 使用專案

---

上一章解說 Sun Java™ Wireless Toolkit for CLDC 如何幫助您進行 MIDP 開發周期。這一章會深入探討使用專案的詳細資訊，包括下列各項：

- 選取專案的目標 API
- 處理 MIDlet 套件屬性，包括 MIDlet 清單
- 瞭解專案目錄結構
- 在專案中包含第三方程式庫

---

### 3.1 選取 API

每個專案都是根據某些 API 的集合建立而成。Sun Java™ Wireless Toolkit for CLDC 可支援多個 API。第 1-7 頁的第 1.6 節「支援的技術」詳細列出了其完整清單。工具組可讓您根據將來要執行軟體的裝置類型，開發某些 API 子集的應用程式。

例如，雖然工具組支援 Mobile 3D Graphics API JSR 184，但您可能想要開發不會利用到該 API 的應用程式。專案的 [API 選擇] 設定可讓您只選擇要使用的 API。

若要瞭解此運作方式，請啟動該工具組並開啓專案。按一下 [設定] 以顯示圖 3-1 中所示的視窗：

圖 3-1 專案設定視窗



在 [API 選擇] 窗格上，[目標平台] 設定控制窗格其他部分的外觀。請選擇最符合需要的設定，並用下列控制項調整選項。例如，如果您要開發用於與 Java Technology for the Wireless Industry JSR 相容裝置的應用程式，請從組合方塊中選擇 [JTWI]。然後使用下方的控制項指定 CLDC 版本並選擇選擇性 API。

當您編譯原始碼時，工具組會套用您的選擇項目。

---

**備註 –** API 選擇項目不適用於模擬器。模擬器總是支援所有可用的 API。您在專案設定中選擇的 API 項目只適用於生成專案。基本上，API 選擇項目是選擇工具組於編譯及預檢驗原始碼檔案時，所用的類別路徑。

---

## 3.2 變更 MIDlet 套件屬性

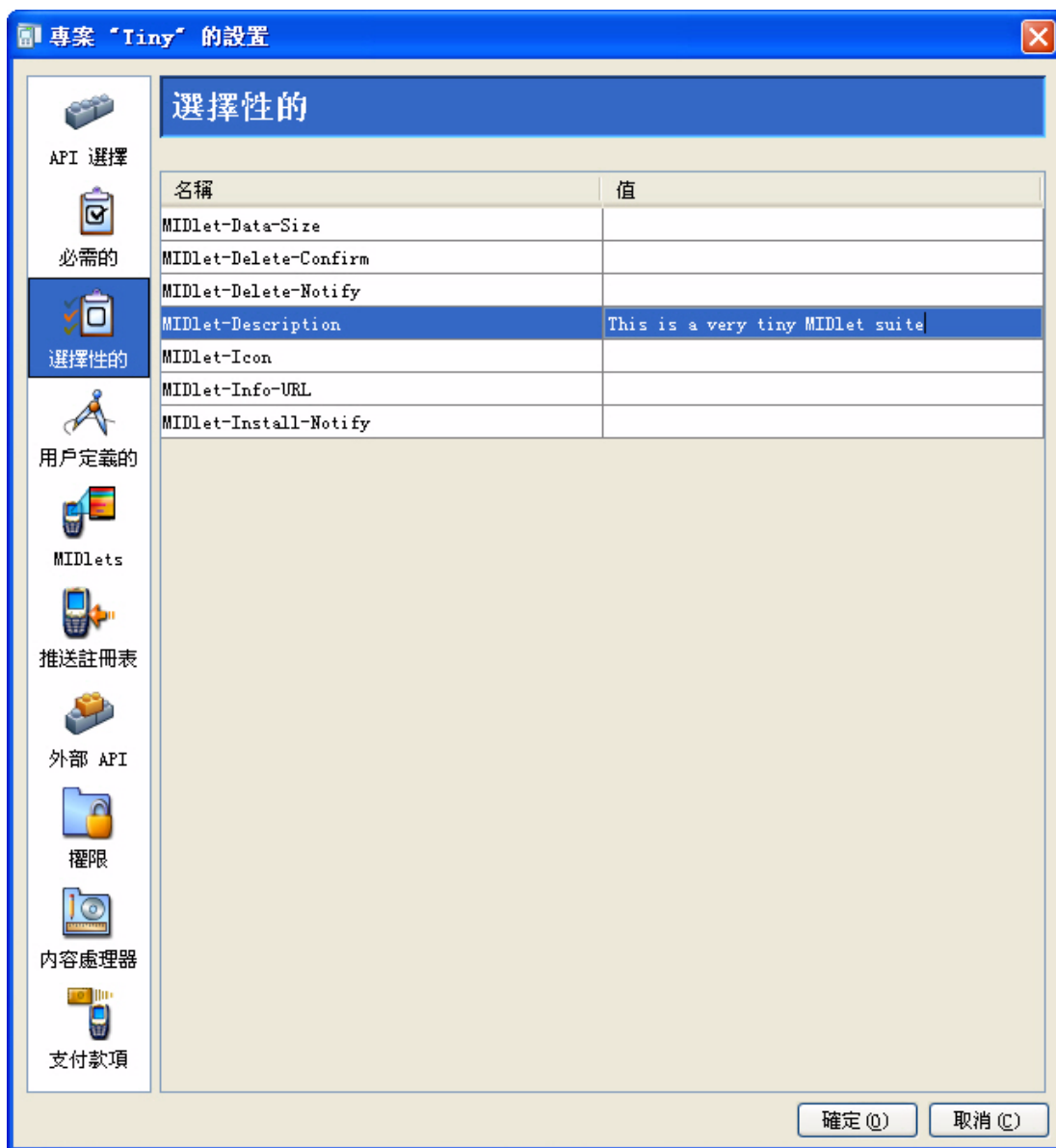
專案設定視窗也可讓您控制 MIDlet 套件屬性，這些屬性儲存在描述元以及 MIDlet 套件 JAR 的清單檔案內。

若要查看屬性，請開啓某專案，然後按一下 [ 設定 ] 按鈕。設定視窗左側的圖示列有三個屬性圖示：[ 必需的 ]、[ 選擇性的 ] 以及 [ 使用者定義 ]。

如需必需的與選擇性的屬性定義，請參閱 MIDP 2.0 規格最終發行版本 2 ( 亦稱爲 MIDP 2.1 )。Sun Java™ Wireless Toolkit for CLDC 會負責處理大部分細節。在開發前期，您可能完全不必顧慮屬性問題。但是當應用程式逐漸穩定，您開始考慮將應用程式部署在實際裝置上並準備上市時，就應該調整這些值。

若要調整 [ 必需的 ] 或 [ 選擇性的 ] 窗格上的值，請在要變更的屬性機碼旁的儲存格中按一下。輸入新的值。

圖 3-2 編輯 MIDlet 套件屬性



若要建立新的使用者定義的屬性，請按一下 [ 用戶定義的 ] 圖示。按一下 [ 增加 ] 按鈕並填入特性名稱和值，然後按一下 [ 確定 ]。

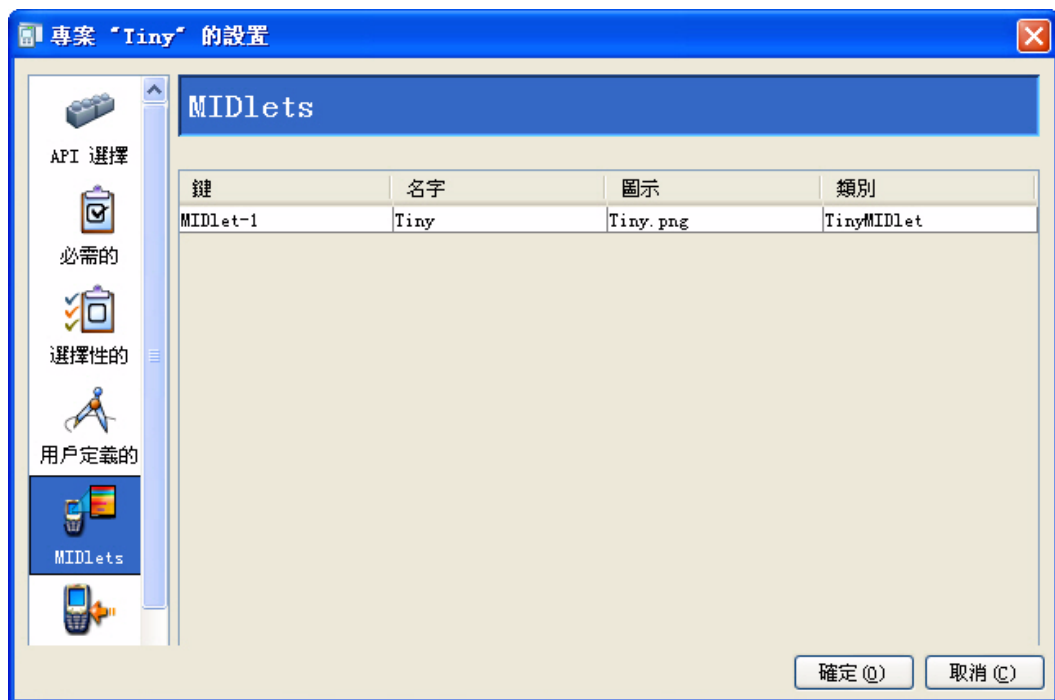
若要編輯使用者定義的特性值，請按一下機碼旁的值欄，以設定必需或選擇性屬性的方式編輯屬性值。

若要移除屬性，請選取屬性，然後按一下 [ 移除 ]。

## 3.3 處理 MIDlet

專案設定同時提供一種方法，可新增或修改包含在當前 MIDlet 套件專案內的 MIDlet。若要瞭解此運作方式，請啟動工具組並開啓現有專案。按一下 [設定]，然後選擇 [MIDlets] 圖示。就會顯示專案中所有 MIDlet 的清單。如果剛建好新專案，工具組會自動填入第一個 MIDlet 項目。

圖 3-3 專案中的 MIDlet 清單



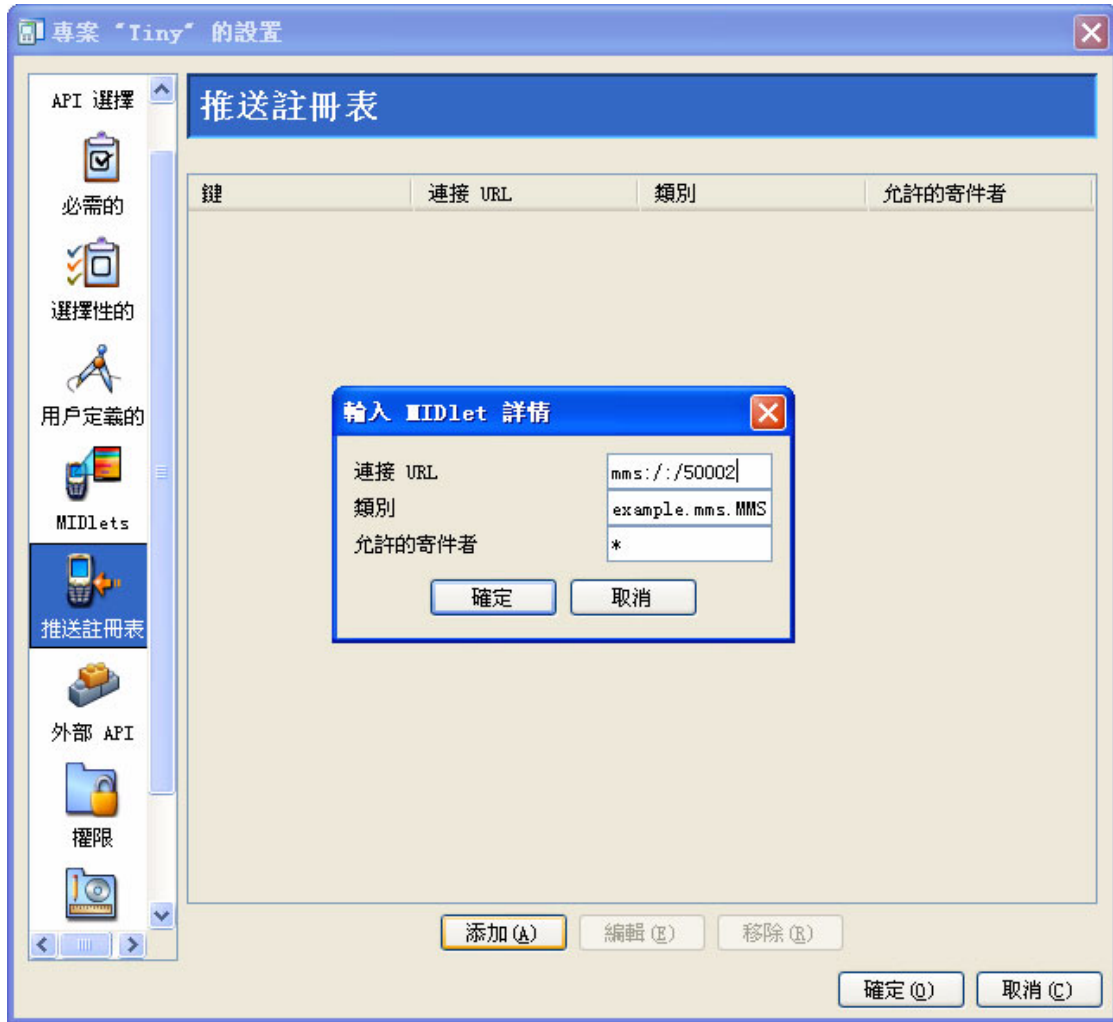
若要增加新的 MIDlet，請按一下 [增加]，然後填入名稱、圖示檔案名稱與類別名稱。圖示檔案名稱可以保持空白。若要變更值或移除 MIDlet 項目，請使用 [編輯] 與 [移除] 按鈕。

使用者看到的 MIDlet 名稱順序是按照 MIDlet 套件的啟動順序。若要修改順序，請選取 MIDlet，再按一下 [上移] 或 [下移]。

## 3.4 使用推送註冊表

您也可以使用專案設定，來處理 MIDlet 套件的推送註冊表設定。請按一下 [ 設定 ... ]，並選擇 [ 推送註冊表 ] 圖示。

圖 3-4 專案推送註冊表設定



若要在推送註冊表中新增項目，請按下 [ 增加 ]，並填入連線 URL、MIDlet 類別及允許的寄件者值，然後按一下 [ 確定 ]。若要編輯某項目，請選取該項目，並按下 [ 編輯 ] 按鈕。若要移除推送註冊表項目，請選取該項目，再按一下 [ 移除 ]。

如果您為應用程式建立推送註冊表項目，請確定要一併輸入適當的權限。如需詳細資訊，請參閱[第 6 章](#)。

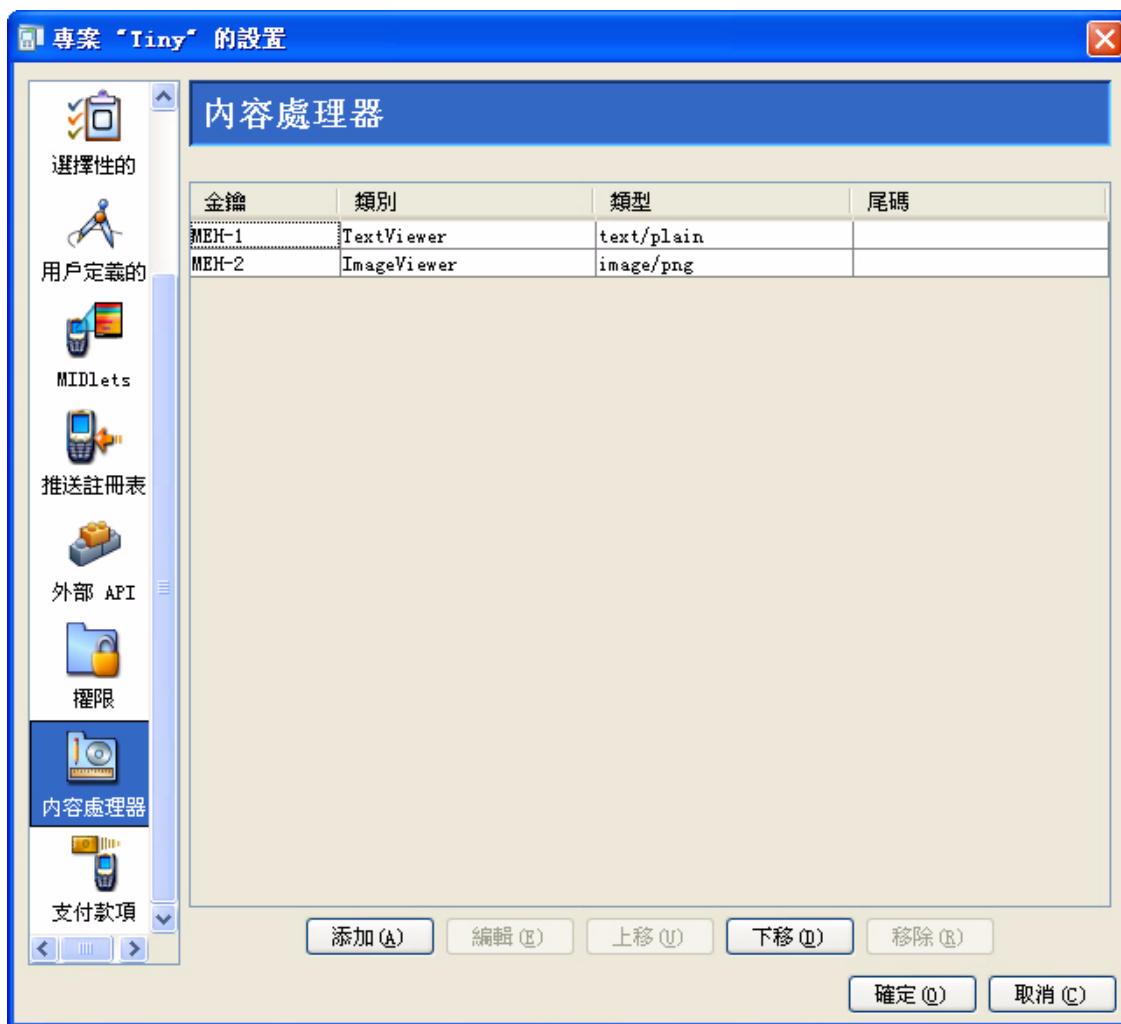
---

## 3.5 設定內容處理器

Sun Java™ Wireless Toolkit for CLDC 可支援由 JSR 211 定義的內容處理器 API (CHAPI)。CHAPI 的基本概念是 MIDlet 可依據內送的内容 ( 檔案 ) 啟動。現代行動電話可使用 SMS、紅外線、藍芽、電子郵件以及其他方式接收内容。大多數内容具有相關聯的内容類型。CHAPI 可指定系統，以便依據指定的内容類型啟動 MIDlet。

若要修改專案中的内容處理器設定，請按一下 [ 設定 ]，然後選擇 [ 内容處理器 ] 窗格。

圖 3-5 配置內容處理器



清單中的每一行表示內容處理器的設定。此範例配置了兩個內容處理器，一個用於 TextViewer，另一個用於 ImageViewer。若要建立新的內容處理器，請按下 [ 增加 ]；若要編輯現有的內容處理器，請按下 [ 編輯 ]。您可以選取一個內容處理器並使用 [ 上移 ] 與 [ 下移 ] 按鈕來調整其順序。若要移除清單中的內容處理器，請按下 [ 移除 ]。



當您增加或編輯內容處理器時，將顯示詳細資訊視窗：

圖 3-6 內容處理器詳細資訊

輸入內容處理器詳細資訊

內容處理器 動作

基本設定

類別: example.text.TextView

ID: example.text.TextView

內容類型

text/plain

添加 (A) 移除 (R)

尾碼字串

添加 (A) 移除 (R)

已允許存取

添加 (A) 移除 (R)

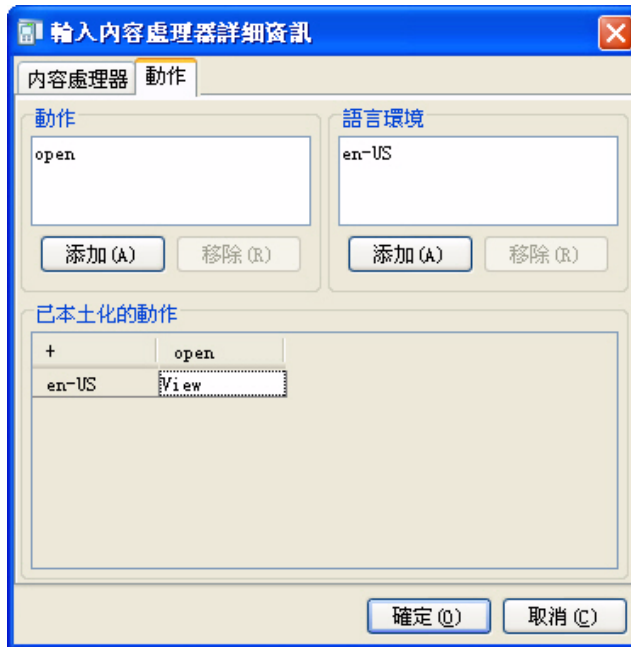
確定 (O) 取消 (C)

在 [ 類別 ] 欄位中填入 MIDlet 的名稱。ID 是用來呼叫內容處理器與控制存取的識別字串。

內容類型是該內容處理器所負責的內容類型清單。使用 [ 增加 ] 或 [ 移除 ] 管理該清單。尾碼字串是 URL 尾碼清單，一般用於代替明確的內容類型。最後，已允許存取是內容處理器的 ID 清單，表示可存取該內容處理器的其他內容處理器。如果該清單為空白，則任何其他內容處理器均可存取該內容處理器。

內容處理器具有相關聯的動作，讓呼叫的應用程式選擇處理內容的方式。例如，影像檢視器內容處理器可能包含檢視影像原始大小的動作，以及讓影像填滿可用螢幕空間的動作。按一下內容處理器詳細資訊視窗的 [ 動作 ] 標籤，以編輯內容處理器的動作。

圖 3-7 內容處理器動作



【動作】清單包含用於此內容處理器之動作的內部名稱。語言環境指顯示易讀動作名稱時，可用的所有語言環境清單。本土化動作是一個表格，其中包含各種語言環境的易讀動作名稱。每一列是一種語言環境，欄則列出各個動作。閱讀一整列便可查看特定語言環境的所有易讀動作名稱。

## 3.6 專案目錄結構

每個專案都有標準的目錄結構。專案本身由 `apps` 子目錄中的目錄代表。例如，在 Windows 上，`demos` 專案位於 `workdir\apps\demos` 中。以下表格描述了專案目錄結構。

表格 3-1 專案目錄結構

目錄	描述
<code>bin</code>	封裝專案時，MIDlet 套件描述元與 JAR 檔案放在此目錄內。此目錄也包含未封裝的清單資訊，而且可能包含使用 [經由 OTA 執行] 時供內部使用的 HTML 檔案。
<code>classes</code>	此目錄用於儲存已編譯的類別檔案。
<code>lib</code>	將第三方程式庫放入此目錄，以將其包含在此專案中。
<code>res</code>	將影像、聲音與其他資源檔案存放於此目錄中。這些檔案會封裝入 MIDlet 套件 JAR 檔案的根目錄。
<code>src</code>	將原始碼檔案放入此目錄。
<code>tmpclasses</code>	由工具組使用。
<code>tmplib</code>	由工具組使用。

此外，專案目錄包含一個內含專案相關資訊的 `project.properties` 檔案。

若要移除專案的暫存目錄與檔案，請選擇 [專案] > [清除]。

## 3.7 使用第三方程式庫

Sun Java™ Wireless Toolkit for CLDC 可讓您在應用程式中納入第三方程式庫。使用第三方程式庫可省去自行生成功能的時間從而縮短開發時間，但是您應該密切注意 MIDlet 套件 JAR 檔案的大小。

在應用程式中使用第三方程式庫時，第三方程式庫的大小會加進您的 JAR 中。您可以使用混淆器縮小程序碼的大小，而且好的混淆器甚至會去除程式庫中您用不到的部分。即便使用混淆器，第三方程式庫的大小仍然可能比您從頭開始仔細撰寫的自訂程式碼大。您必須權衡在縮短開發時間與 MIDlet 套件 JAR 檔案的大小之間如何取捨。

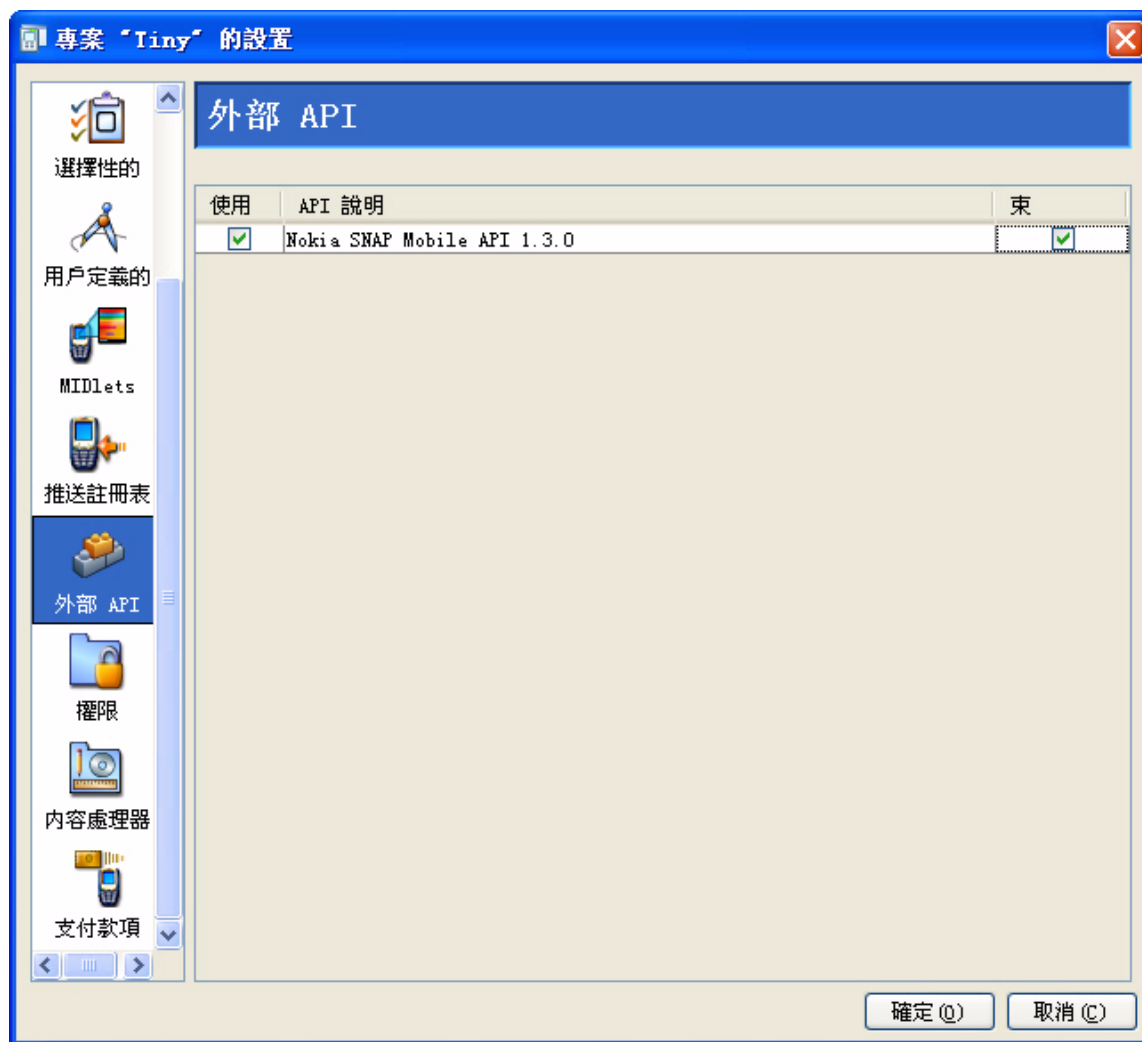
Sun Java™ Wireless Toolkit for CLDC 提供兩種併入第三方程式庫的方式。專案設定中的 [外部 API] 窗格可讓您輕鬆在專案中包含或排除程式庫。另外，您可以將程式庫置於特定位置，讓一個或所有專案得以使用。

## 3.7.1 使用外部 API

若要指定應包含的 API，請按一下 [ 設定 ]，然後按一下 [ 外部 API ] 圖示。系統將顯示可用的外部 API 清單（此清單可能為空白）。

若要在生成時在類別路徑中增加 API，請核取 [ 使用 ] 欄中的方塊（請參閱圖 3-8）。如果您想要 API 隨附至應用程式，還需核取 [ 束 ] 方塊。如果您知道應用程式要部署於缺少所選外部 API 的裝置上，請將這些 API 隨附於您的應用程式。

圖 3-8 選擇外部 API



## 3.7.2 一個專案的第三方程式庫

放在專案 `lib` 目錄內的任何程式庫檔案，都會包含在專案的生成與封裝中。程式庫應該是 Java 技術類別的 JAR 或 Zip 檔案。

因此，如果您安裝在 Windows 上且應用程式名為 Tiny，則類別程式庫將位於 `workdir\apps\Tiny\lib`。當您生成、執行、除錯與封裝專案時，會使用 `lib` 目錄中的類別檔案。

## 3.7.3 所有專案的第三方程式庫

有些裝置的程式庫可供所有安裝的 MIDlet 套件使用。例如，某家製造商可能提供額外的 API，讓該公司的所有裝置都可以使用。在這種情況下，您會希望在生成與測試應用程式時，能夠使用這些程式庫。但是因為安裝 MIDlet 套件的裝置上已經存在該程式庫，所以您不希望將程式庫包含在您的封裝 MIDlet 套件中。

只要將程式庫放在 `workdir\apps\lib` 目錄中即可實現此功能。此目錄中的程式庫可供所有專案使用。

---

## 3.8 配置 Wireless Toolkit

工具組包括一些進階的配置選項。您可透過編輯 `ktools.properties` 檔案的副本使用這些選項，該檔案可在以下位置找到：

---

**Windows:**    `toolkit\wtklib\Windows\ktools.properties`

**Linux:**        `toolkit/wtklib/Linux/ktools.properties`

---

將 `ktools.properties` 複製到 `workdir/os/wtklib`，並按本小節其餘部分的描述進行變更。

- [變更主控台字型](#)
- [設定應用程式目錄](#)
- [設定 javac 編碼特性](#)
- [使用修正控制系統](#)

變更 `ktools.properties` 所產生的效果在下次啟動工具組時才能看到。

### 3.8.1 變更主控台字型

若要變更主控台（和其他文字區域）中使用的字型，可以編輯兩個特性。以下為將字型變更爲 Times New Roman 20 號的範例：

```
font.JTextArea=Times New Roman  
font.size.JTextArea=20
```

若要復原預設字型與大小，請移除這兩個特性。

### 3.8.2 設定應用程式目錄

依預設，Sun Java™ Wireless Toolkit for CLDC 將應用程式儲存在您工作目錄的 `apps` 子目錄中。您可以在 `ktools.properties` 中以下列格式新增一行來做變更：

`kvem.apps.dir:` 應用程式目錄

對於 Windows，目錄路徑中任何反斜線 (\) 字元之前，必須加上另一條反斜線。而且，目錄路徑不得包含任何空格。

例如，若要將應用程式目錄設定爲 `D:\dev\midlets`，請使用：

```
kvem.apps.dir: D:\\dev\\midlets
```

Linux 路徑則可以依常規指定。

### 3.8.3 設定 javac 編碼特性

依預設，`javac` 使用您正在執行之 Java SE 平台所設定的編碼。如需有關如何置換預設原始碼檔案編碼的資訊，請參閱[附錄 C](#)。

### 3.8.4 使用修正控制系統

使用 `filterRevisionControl` 特性可以配置工具組，以辨識並忽略 SCCS、RCS 與 CVS 修正控制系統所建立的輔助檔案。

若要識別並忽略輔助檔案，請在 `ktools.properties` 中包括以下行：

```
kvem.filterRevisionControl: true
```

這樣可以防止工具組將修正控制檔案視爲原始碼檔案和資源檔案。例如，工具組會將名稱爲 `src\SCCS\s.MyClass.java` 的檔案視爲 SCCS 修正控制檔案，而非 Java 技術原始碼檔案。

# 使用模擬器

Sun Java™ Wireless Toolkit for CLDC 模擬器在桌上型電腦上模擬 MIDP 裝置。這種作法，可以方便您觀察應用程式在 MIDP 環境中的運作方式，還能讓開發周期更為緊湊，而且完全包含在桌上型電腦上。

模擬器不代表特定裝置，但可以正確實作它所支援的 API。

## 4.1 模擬器面板

面板是模擬器實作最頂端的一層，可賦予模擬器某種外觀、螢幕特性和輸入控制。Sun Java™ Wireless Toolkit for CLDC 隨附一些代表不同裝置種類的面板。

表格 4-1 模擬器面板

名稱	螢幕大小	畫布大小	色彩	輸入
DefaultColorPhone	240 x 320	240 x 289	4096	ITU-T
DefaultGrayPhone	180 x 208	180 x 177	4096	ITU-T
MediaControlSkin	180 x 208	180 x 177	4096	ITU-T
QwertyDevice	636 x 235	540 x 204	4096	Qwerty

您可以任意建立您自己的模擬器面板。如需詳細資訊，請參閱「基本自訂指南」。

## 4.2 模擬器控制

模擬器在標準桌上型電腦視窗中的外觀與運作方式類似行動電話。本小節說明如何控制模擬器。雖然說明與圖形以 DefaultColorPhone 面板為基礎，但所有面板的運作方式相類似。

圖 4-1 DefaultColorPhone 模擬器面板





您可以使用滑鼠按一下按鈕以按下它們。大部分按鈕還有通常使用起來更為便捷的鍵盤捷徑。鍵盤數字 0 到 9 對應模擬器的 0 到 9 按鈕。下表列出比較不明顯的鍵盤捷徑。

表格 4-2      鍵盤捷徑

模擬器按鈕	鍵盤鍵
左邊軟式按鈕	F1 鍵
右邊軟式按鈕	F2 鍵
電源按鈕	Esc
選取	Enter

文字的輸入方式與許多實際裝置所使用的輸入方式類似。多按幾次數字按鈕可得到您要的字母。例如，按兩次按鍵 5 即可輸入字母 K。輸入文字時，星號鍵 (\*) 可切換大寫、小寫、數字與符號。畫面最上面的指示器可顯示目前的模式。井字鍵 (#) 可用於輸入空格。

此外，您可以直接在鍵盤上輸入文字。雖然如此便於輸入文字，但您須謹記，使用者很可能沒有這麼方便的輸入法。

另一個便利之處在於可在文字區複製與貼上資訊。同時按下 Ctrl+V，可將剪貼簿中的文字貼至 TextBox 或 TextField。若要複製 TextBox 或 TextField 中的內容，請同時按下 Ctrl+C。此操作可將文字欄位的完整內容置於剪貼簿中。

## 4.3      設定模擬器偏好

您可以調整模擬器設定，以更加類似特定裝置，或在不同資源條件下測試您的應用程式。

### 4.3.1      網路代理伺服器

模擬器使用桌上型電腦的網路連線。例如，當模擬器執行會建立 HTTP 連線的 MIDlet 時，模擬器會嘗試使用桌上型電腦的網路設定來建立 HTTP 連線。

如果開發電腦位於防火牆後方，您可以使用代理伺服器建立 HTTP 連線。如果不確定，請檢查瀏覽器的設定以查看它是否使用代理伺服器。

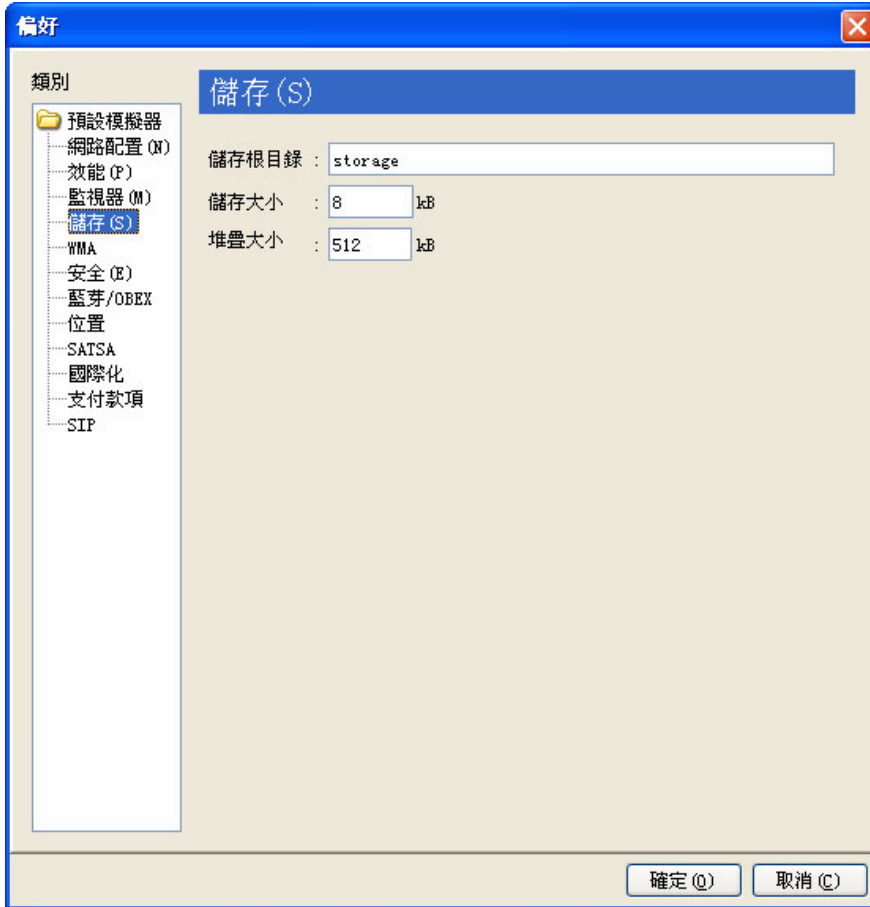
如果您使用代理伺服器，您必須將模擬器配置成使用同一部代理伺服器。若要執行此操作，請選擇 [編輯] > [偏好]。在 [網路配置] 窗格上，填入要使用的代理伺服器名稱與連接埠號。您也可以選取要使用的 HTTP 版本。

如果您的代理伺服器使用 HTTP 基本認證 (請參閱 RFC 2617)，請核取 [認證] 並填入使用者名稱與密碼。

## 4.3.2 儲存容量大小

您可以設定或修改指定給每個模擬器的永久性儲存裝置的容量，也可以變更堆疊大小（應用程式物件所容許的儲存容量）。選擇 [編輯] > [偏好]，然後選取 [儲存] 項目。

圖 4-2 儲存偏好



### 4.3.2.1 永久性儲存裝置

模擬器具有永久性儲存裝置，其預設位置為電話面板目錄中的 `appdb` 子目錄下。這些檔案的副檔名為 `.db`。

例如，在 Windows 上，DefaultColorPhone 模擬器面板的永久性儲存裝置儲存在 `workdir\appdb\DefaultColorPhone\manager_storage_settings.db` 中。

如果同時執行同一個模擬器面板的多個實例，工具組會為每個實例產生唯一的檔案路徑。例如，在 Windows 上，DefaultColorPhone 實例的檔案路徑名稱可能為 `workdir\appdb\temp.DefaultColorPhone1`、`workdir\appdb\temp.DefaultColorPhone2` 等等。

---

**備註** – 檔案 `workdir\appdb\DefaultColorPhone\in.use` 會追蹤標記為使用中的儲存根目錄數目。如果模擬器當機，則您需要刪除 `in.use` 檔案。

---

本工具組可讓您選擇不同的檔案儲存位置，而且您可以限制儲存裝置的大小。當可用的永久性儲存裝置很小時，這對於測試應用程式的運作方式很有用。

若要調整永久性儲存裝置設定，請選擇 [編輯] > [偏好]，再按一下左側窗格中的 [儲存]。在 [儲存根目錄] 欄位中，輸入您希望用於永久性儲存裝置的目錄名稱。您只可以輸入相對路徑，而且您指定的目錄會建立在 `appdb` 子目錄中。

依預設，您可以使用大小為 1MB (1024KB) 的永久性儲存裝置。您可以以 KB 為單位輸入上限。請記住，除應用程式使用空間外，儲存裝置實作也會耗用一些空間。例如，如果永久性儲存裝置的大小是 8KB，則應用程式資料和儲存裝置耗用空間會共用這 8192 個位元組。

若要清除模擬器的永久性儲存裝置，請選擇 [檔案] > [公用程式]。按一下 [清除資料庫] 按鈕即可清除永久性儲存裝置。清除資料庫不會影響安裝的應用程式。

### 4.3.2.2 堆疊大小

堆疊是用於儲存應用程式物件的記憶體。若要變更堆疊大小，請選擇 [編輯] > [偏好]，然後選取 [儲存] 項目 (請參閱圖 4-2)。依預設，堆疊大小為 1MB。您可以設定最大堆疊大小，以便更真切模擬實際裝置上的狀況。在 [堆疊大小] 欄位中填入最大堆疊大小 (以 KB 為單位)。

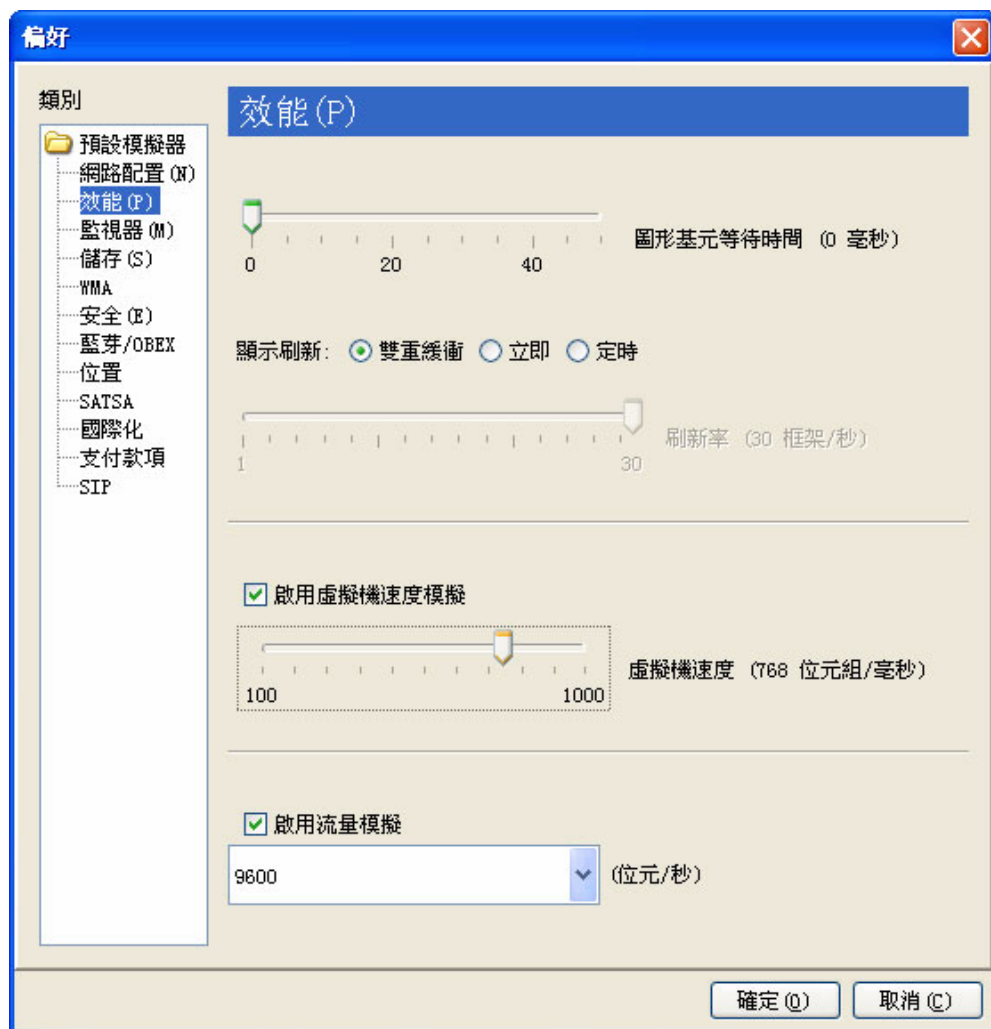
## 4.3.3 調整模擬器效能

模擬器使用桌上型電腦的許多資源，包括顯示器與網路連線。與桌上型模擬器相比，實際的 MIDP 裝置通常配備的處理器比較慢、記憶體比較少、網路連線也比較慢，而且顯示器類型也可能不同。

Sun Java™ Wireless Toolkit for CLDC 可讓您模擬實際裝置受限的環境。雖然模擬器不代表實際裝置，但是調整效能設定將為您提供不同執行階段下應用程式的效能資訊。

選擇 [編輯] > [偏好]，然後在左側窗格中按一下 [效能]。請參閱圖 4-3。

圖 4-3 調整模擬器效能



調整【圖形基元等待時間】可改變應用程式呼叫 Graphics 類別中的繪圖方法後，至繪圖行為實際發生所經過的時間。

若要變更螢幕特性，請選擇【顯示器刷新】類型之一。如果選擇【定時】類型，也必須指定【刷新率】。

若要模擬速度較慢的實際裝置，請核取【啟用虛擬機器速度模擬】，並選擇所要的速度。

若要調整模擬網路速度，請核取【啟用流量模擬】，然後以位元 / 秒為單位選擇速度。

---

## 4.4 暫停與繼續

MIDlet 有 MIDP 規格所定義的生命周期。MIDlet 可以由裝置啟動及停止。此外，外部事件（例如來電）可能導致裝置將 MIDlet 暫停。

模擬器提供一個簡單的機制，可暫停及重新繼續正在執行的 MIDlet。這對於測試應用程式暫停時的運作方式非常有用。

當模擬器正在執行時，從模擬器的視窗功能表中，選擇 [MIDlet] > [ 暫停 ]。執行中的 MIDlet 會暫停，而且螢幕顯示「Incoming Call...」訊息。

若要重新繼續作業，請從功能表選擇 [MIDlet] > [ 重新繼續 ]。

---

## 4.5 單獨執行模擬器

在開發過程中，通常您會直接在工具組中按下 [ 執行 ] 按鈕或選取 [ 專案 ] > [ 經由 OTA 執行 ] 功能，來執行模擬器。可是如果要進行測試或示範，您可能想單獨執行模擬器本身。這一節會說明幾種不同方法。由 Sun Java™ Wireless Toolkit for CLDC 安裝程式建立的程式群組包含幾個與模擬器單獨執行有關的項目。

- 若要直接執行應用程式（類似於按下 [ 執行 ] 按鈕），請選擇 [ 執行 MIDP 應用程式 ... ] 項目。工具組會提示您找出本機磁碟上的 MIDlet 描述元檔案。請注意，對應的 MIDlet 套件 JAR 也必須存在。
- 若要執行模擬器的應用程式管理軟體 (AMS)，請選擇 [OTA 佈建] 項目（大致類似於使用者介面中的 [ 經由 OTA 執行 ] 功能）。模擬器會顯示 AMS 歡迎畫面，而且您可藉由輸入 URL 的方式安裝應用程式。
- 若要變更模擬器的偏好，請從工具組的程式群組中選擇 [ 偏好 ] 項目。所開啓的偏好視窗，等同於從使用者介面中選擇 [ 編輯 ] > [ 偏好 ... ] 時出現的視窗。
- 不執行使用者介面也可以存取 Sun Java™ Wireless Toolkit for CLDC 公用程式。只要選擇 [ 公用程式 ] 項目即可。
- 最後，您可以變更預設使用的模擬器面板。請選擇 [ 預設裝置選項 ] 項目，並選擇其中一個可用的模擬器面板。下次啟動模擬器時，就會使用選取的面板。

您也可以從指令提示符號執行模擬器。如需詳細資訊，請參閱[附錄 B](#)。

---

## 4.6 使用第三方模擬器

第三方公司（例如裝置製造商或無線電信業者）有時候會建立與工具組相容的裝置模擬器。您可以在工具組中安裝其他模擬器，以取得在範圍廣泛的實作上執行應用程式的經驗。進行程序通常是解壓縮或安裝協力廠商模擬器，然後將其目錄複製到 `workdir\wtklib\devices` 中。下次執行工具組，就可以使用該模擬器。

您可以在下列網址找到目前可用模擬器的部分清單：

<http://developers.sun.com/techtopics/mobility/midp/articles/emulators/>

## 監視應用程式

---

Sun Java™ Wireless Toolkit for CLDC 提供幾種工具，可以監視應用程式的運作方式。這些工具在除錯與最佳化程式碼時，非常有用：

- 效能評測器會列出應用程式中，每一種方法的使用次數與執行時間。
- 記憶體監視器會在應用程式執行時，顯示記憶體使用率。
- 網路監視器會顯示應用程式傳送與接收的網路資料。它支援許多網路協定，包括 HTTP、HTTPS、SMS 與 CBS。
- 追蹤會在工具組主控台上顯示低階資訊。

---

**備註** – 監視功能可能會降低應用程式的執行速度。

---

---

### 5.1 使用效能評測器

效能評測器會追蹤應用程式中的每一種方法。對於特定執行，它會指出每一種方法花費的時間長度，以及每一種方法的呼叫次數。當應用程式執行完畢，模擬器關閉之後，就會顯示效能評測器，讓您可以瀏覽方法呼叫資訊。

若要開啓效能評測器，請選擇 [編輯] > [偏好]。請參閱圖 5-1。在左側窗格中按一下 [監視器]。在右側窗格中，請核取 [啟動效能評測]。如果想要看到所有系統實作方法的效能評測資訊，請核取 [顯示系統類別]。否則，效能評測器只會顯示內含呼叫您的應用程式方法的系統方法。按一下 [確定]。

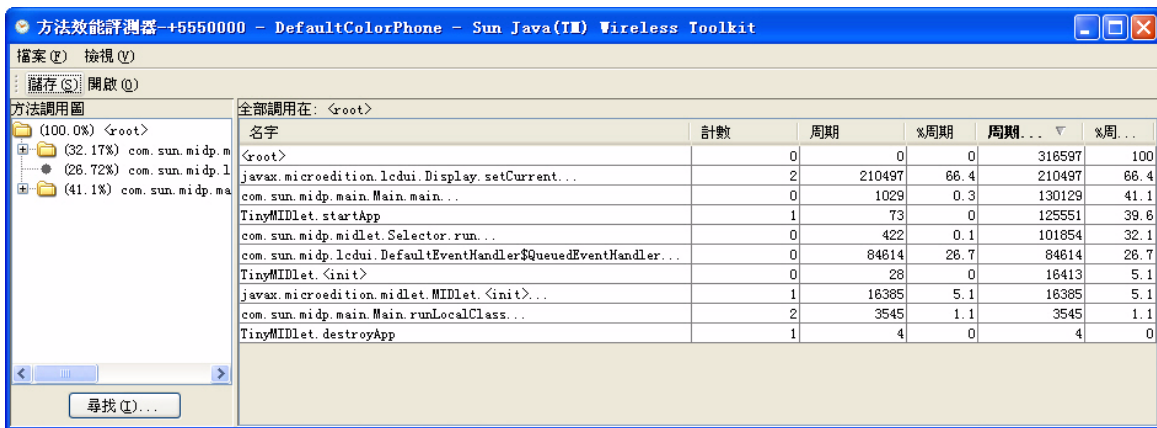
圖 5-1 開啟效能評測器



現在，請按一下【執行】按鈕啟動您的應用程式。依照正常方式，與應用程式互動。完成時，關閉模擬器。就會顯示效能評測器，列出應用程式中呼叫的所有方法的相關資訊。



圖 5-2 方法效能評測器



效能評測器會顯示二種資訊類型：

- 以階層式清單顯示的方法關係叫作「調用圖」。
- 效能評測器右側顯示每個方法及其子項的執行時間與呼叫次數。

**備註** – 從模擬器得來的效能評測值不代表在實際裝置上的實際值。

## 5.1.1 調用圖

調用圖顯示方法呼叫的階層。呼叫其他方法的方法會顯示為資料夾。連按兩下某方法，可開啓該方法，並查看它呼叫的方法。不呼叫其他任何方法的方法會顯示成灰色圓形。

您可以搜尋特定類別或方法名稱。請按一下 [ 尋找 ]，並填入名稱。就會從調用圖中目前的選項開始執行搜尋，直到圖形末端。如果要搜尋整個調用圖，請先核取 [ 環繞 ]，再按一下 [ 尋找 ] 按鈕。

按調用圖的不同節點時，效能評測器右側會顯示該節點的方法詳細資訊。

## 5.1.2 執行時間與呼叫次數

效能評測器視窗右側會顯示有關方法的詳細資訊。您可以看到方法名稱、呼叫次數及模擬器用於執行該方法的時間長度。執行時間以 4 種不同方式描述：

- [ 周期 ] 顯示處理器執行方法本身時所花費的時間。
- [% 周期 ] 是指花費在方法本身的總執行時間百分比。
- [ 周期 ( 含子類 ) ] 是花費在方法以及它所呼叫的方法的時間。
- [% 周期 ( 含子類 ) ] 顯示花費在方法及其呼叫方法的時間，與總執行時間的比值。

按一下任何欄，可依該欄排序。再按一次，可切換依遞增或遞減順序排序。

右側窗格顯示調用圖中，目前選取節點所包含的方法。如果要查看每一種方法，請按一下調用圖的 <root> 節點。

## 5.1.3 儲存與載入效能評測器資訊

若要儲存效能評測器階段作業，請在效能評測器視窗中，按一下 [ 儲存 ] 按鈕。選擇檔案名稱。

若要載入效能評測器階段作業，請選擇 [ 檔案 ] > [ 公用程式 ]。按一下 [ 效能評測器 ]，然後按下 [ 啟動 ]。選取檔案後，就會顯示效能評測器視窗，列出所有階段作業資訊。

---

## 5.2 使用記憶體監視器

許多 MIDP 裝置的記憶體都有限。Sun Java™ Wireless Toolkit for CLDC 包含一個記憶體監視器，方便您檢查應用程式的記憶體使用率。您可以查看應用程式使用的總記憶體，以及各物件的記憶體使用率詳細清單。

若要開啓記憶體監視器，請選擇 [ 編輯 ] > [ 偏好 ]。在左側窗格中按一下 [ 監視器 ]。核取 [ 啓用記憶體監視器 ]。

下次執行模擬器時，就會出現記憶體監視器視窗，顯示應用程式長期的記憶體使用率圖形。記憶體監視器會降低應用程式啟動時的速度，因為必須記錄每一個建立的物件。

圖 5-3 記憶體監視器圖形



記憶體監視器圖形會顯示下列資訊：

- 目前 — 應用程式目前使用的記憶體數量。
- 最大 — 程式開始執行之後使用的最大記憶體數量，在圖中以紅色虛線顯示。
- 物件 — 堆疊中的物件數。
- 已用 — 已使用的記憶體數量。
- 空閒 — 可用但未使用的記憶體數量。
- 總數 — 啟動時可用的記憶體總數量。

請記住，若要修改堆疊大小，請選取 [ 編輯 ] > [ 偏好 ]，然後選擇 [ 儲存 ] 標籤。如需詳細資訊，請參閱第 3 章。

若要要求系統執行資源回收，請按一下 [ 執行 GC ]。

---

**備註** — 您用模擬器觀察到的記憶體使用率，不會與實際裝置上的記憶體使用率完全一樣。請記住，模擬器不代表實際裝置。而只是它支援的 API 一種可能的實作方式而已。

---

若要查看應用程式中各個物件的詳細資訊，請在記憶體監視器視窗中，按一下 [ 物件 ] 標籤。

圖 5-4 記憶體監視器物件顯示

名字	生存	合計	合計大小	平均大小
虛擬機內部	50	60	6872	137
java.lang.OutOfMemoryError	1	1	20	20
java.lang.String[ ]	32	81	1300	40
java.lang.Thread	3	3	84	28
char[ ]	690	15617	50080	72
java.io.PrintStream	1	1	28	28
com.sun.midp.io.SystemOutputStream	1	1	12	12
java.io.OutputStreamWriter	1	1	28	28
java.lang.String	418	1489	10032	24
java.lang.StringBuffer	31	212	744	24
com.sun.clde.i18n.uclc.DefaultCaseConverter	1	1	12	12

類別: 1859      已用: 93172 位元組      空閒: 431116 位元組      合計: 524288 位元組

將顯示具有以下欄的表格：

- 名稱 — 物件的類別名稱。
- 生存 — 實例數。這些實例中，有些可能符合資源回收的條件。
- 總數 — 應用程式開始執行之後已配置的物件總數。
- 合計大小 — 物件目前使用的記憶體數量。
- 平均大小 — 平均物件大小，計算方式為總容量除以生存實例。

按一下任何欄標頭，可依該欄排序。

您可以從記憶體監視器視窗功能表選擇 [ 檢視 ] > [ 尋找 ... ]，以搜尋特定類別名稱。

## 5.2.1 儲存與載入記憶體監視器資訊

若要儲存記憶體監視器階段作業，請按一下 [ 儲存 ] 按鈕。選擇檔案名稱。

若要載入記憶體監視器階段作業，請選擇 [ 檔案 ] > [ 公用程式 ]。按一下 [ 記憶體監視器 ]，然後按下 [ 啟動 ]。選取檔案後，就會顯示記憶體監視器視窗，列出所有階段作業資訊。

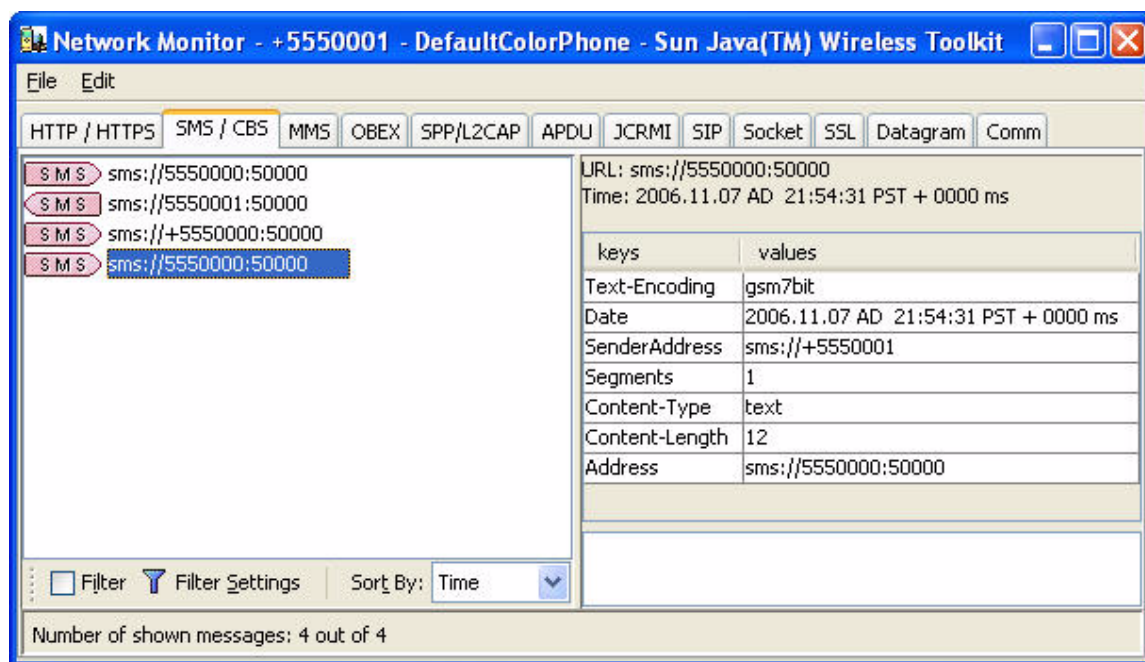
## 5.3 使用網路監視器

MIDP 應用程式至少都具有 HTTP 網路連線功能，但還是可能使用其他類型的網路連線。網路監視器提供方便的方法，讓您查看應用程式在網路上傳送與接收的資訊。如果要進行網路連線除錯，或找出最佳化網路流量的方法，這一點相當有用。

若要開啓網路監視器，請選擇 [ 編輯 ] > [ 偏好 ]。在左側窗格中按一下 [ 監視器 ]。核取 [ 啓用網路監視 ]。

下次執行模擬器時，就會出現網路監視器視窗。

■ 5-5 網路監視器



當應用程式建立任何類型的網路連線時，都會擷取並顯示有關該連線的資訊。圖中顯示 HTTP 請求與回應。

左側的顯示會列出訊息與訊息片段的階層。按一下訊息或部分訊息，可在網路監視器右側查看詳細資訊。連按兩下訊息或部分訊息以將其展開或摺疊。

訊息內文以原始十六進制值及其對等值顯示。

**備註** – 您可以檢查仍在傳送處理中的訊息。不完整訊息在訊息樹狀結構中以粗體反白顯示。

## 5.3.1 篩選訊息

篩選器對於檢查網路總流量的部分子集相當有用。使用的網路協定不同，篩選器設定也會不同。

按下 [ 篩選器設定 ] 按鈕即可使用篩選器。請依據您的需要變更篩選器設定。

表格 5-1 網路監視器篩選器設定

網路協定	篩選器設定
HTTP/HTTPS	輸入文字以比對 HTTP 訊息的各個部分：URL、狀態行、標頭或內文。例如，在 [URL] 欄位中輸入 slashdot，將只篩選出其 URL 包含 slashdot 的訊息。
SMS/CBS	您可以指定協定、訊息類型與比對方向。此外，您可以輸入文字，以比對寄件者、收件者及訊息內容。
MMS	輸入文字，以比對方向、寄件者、收件者以及副本 / (cc) 和密件副本 (bcc) 收件者。此外，您可以篩選主旨、內容 ID、內容位置、MIME 類型與編碼。
OBEX、SPP/L2CAP	您可以使用 URL 或標頭內容進行篩選。
APDU、JCRMI	篩選 URL 或訊息內容。
SIP	沒有
通訊端、SSL、資料封包、通訊	輸入文字，以比對連線字串 (URL) 或內容。

輸入篩選字串後，按下 [ 確定 ] 回到網路監視器。這時將核取 [ 篩選器 ] 核取方塊，表示正在使用篩選器。若要停用篩選器，查看所有訊息，請取消核取此核取方塊。

## 5.3.2 排序訊息

若要依特定順序排列訊息樹狀結構，請按一下 [ 排序依據 ] 組合方塊，並選擇條件。

- 時間 — 訊息依傳送或接收的時序排序。
- URL — 訊息依 URL 位址排序。相同位址的多個訊息依時間排序。
- 連線 — 訊息依通訊連線排序。使用相同連線的訊息依時間排序。此排序類型可讓您查看依請求及其相關回應分組的訊息。
- 排序參數是依您選擇的訊息協定而定。例如，依時間排序就與通訊端訊息無關。

### 5.3.3 儲存與載入網路監視器資訊

若要儲存網路監視器階段作業，請從網路監視器視窗功能表選擇 [ 檔案 ] > [ 儲存 ] 或 [ 檔案 ] > [ 另存新檔 ]。選擇檔案名稱。

若要載入網路監視器階段作業，請選擇 [ 檔案 ] > [ 公用程式 ]。從清單中選取 [ 網路監視器 ]，然後按下 [ 啟動 ]。選取檔案後，就會顯示網路監視器視窗，列出所有階段作業資訊。

### 5.3.4 清除訊息樹狀結構

若要移除網路監視器上的所有訊息，請從網路監視器視窗功能表選擇 [ 編輯 ] > [ 清除 ]。





## 安全性與 MIDlet 簽署

---

MIDP 2.0 (JSR 118) 包含以保護網域為基礎的全面性安全模型。MIDlet 套件安裝在保護網域中，並由保護網域來決定保護功能的存取權。MIDP 2.0 規格也包含使用公開金鑰加密，來檢驗與驗證 MIDlet 套件的建議方式。

如需確定的資訊，請參閱 MIDP 2.0 規格。如需使用 Sun Java™ Wireless Toolkit for CLDC 的 MIDlet 簽署簡介，請閱讀「Understanding MIDP 2.0's Security Architecture」文章，該文章可在 <http://developers.sun.com/techttopics/mobility/midp/articles/permissions/> 上取得

如需有關公開金鑰加密的詳細背景資料，請閱讀「MIDP Application Security 1: Design Concerns and Cryptography」文章，該文章可在 <http://developers.sun.com/techttopics/mobility/midp/articles/security1/> 上取得

這一章描述 Sun Java™ Wireless Toolkit for CLDC 中保護網域、權限與 MIDlet 簽署的支援。

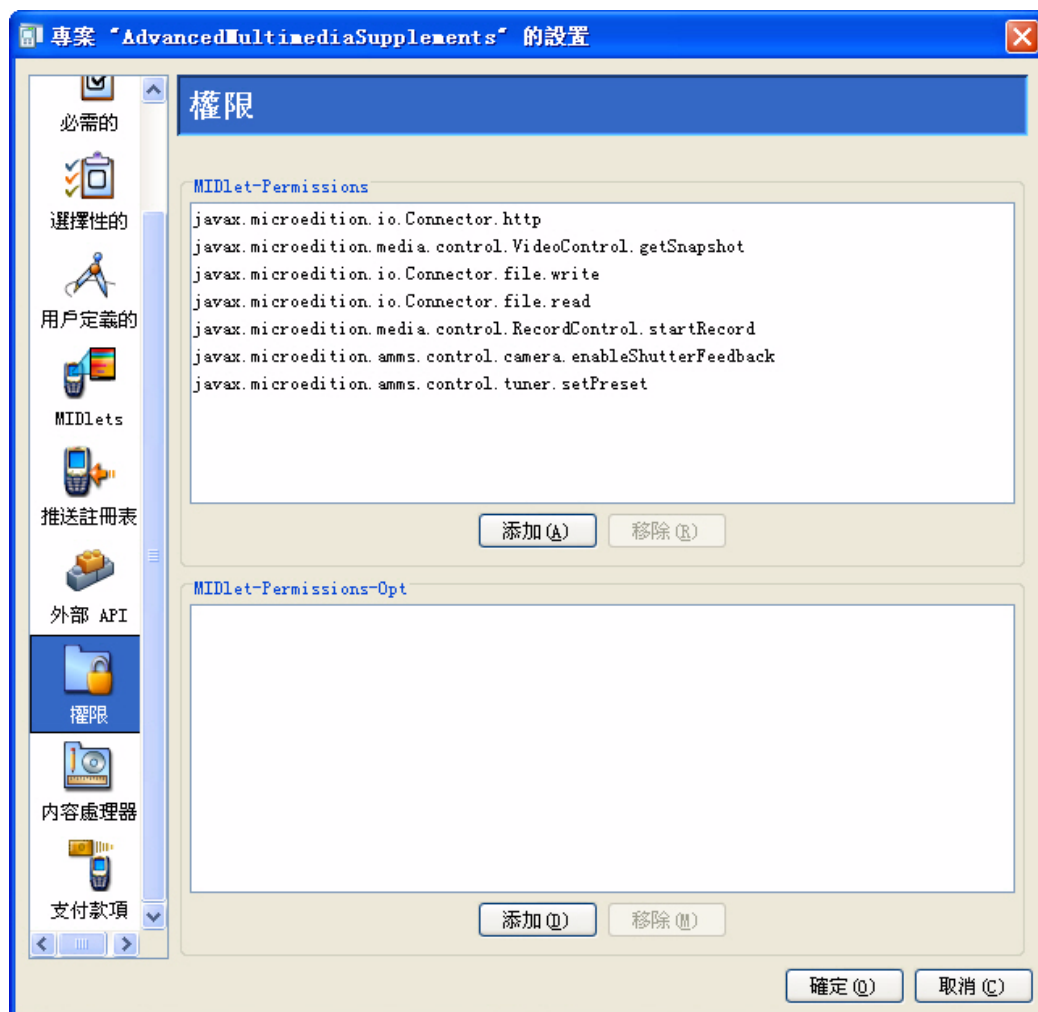
---

### 6.1 權限

MIDlet 必須擁有執行機密性操作的權限，例如與網路連線。權限有特定的名稱，而且 MIDlet 套件可以透過 MIDlet 套件描述元中的屬性，指示是否需要某些權限種類。

您可以按一下 [ 設定 ] 按鈕，將這些權限屬性加入專案中。選取 [ 權限 ] 圖示。[MIDlet-Permissions] 方塊會顯示 MIDlet 必須擁有的權限，而 [MIDlet-Permissions-Opt] 方塊則包含可選的權限。

圖 6-1 MIDlet 套件權限



若要在任一個方塊中新增權限，請按一下 [ 增加 ]，並選擇要新增的權限。若要移除權限，請反白顯示該權限，再按一下 [ 移除 ]。

## 6.2 選取安全性策略

Sun Java™ Wireless Toolkit for CLDC 支援由 JSR 185 (Java Technology for the Wireless Industry) 和 JSR 248 (Mobile Service Architecture, 簡稱 MSA) 定義的安全性策略。在[第 6-3 頁的第 6.2.1 節「MSA 保護網域」](#)和[第 6-4 頁的第 6.2.2 節「Java for the Wireless Toolkit Industry 保護網域」](#)詳細描述了保護網域。

若要選擇您想要模擬器使用的安全性策略，請選取 [ 編輯 ] > [ 偏好 ]，並在 [ 種類 ] 清單中選取 [ 安全 ]。從 [ 安全性策略 ] 組合方塊中，選擇 [MSA] 或 [JTWI]。選取一個可用的安全性策略。

當您使用 [ 經由 OTA 執行 ] 時，您的封裝 MIDlet 套件直接安裝在模擬器中，而且放置在安裝時的保護網域中。模擬器使用公開金鑰加密，來決定安裝 MIDlet 套件的保護網域。

如果 MIDlet 套件未經過簽署，就會放置在預設保護網域。MSA 和 JTWI 的預設值不相同。請參閱 [6.2.1 小節](#)和 [6.2.2 小節](#)。如果 MIDlet 已經過簽署，則放置在與簽署金鑰憑證鏈的根憑證相關的保護網域。

例如，有一家假想的 Respectable Software 公司，想要發行加密簽署的 MIDlet 套件。Respectable Software 向 Super-Trustee (一家假想的憑證授權單位) 購買簽署金鑰組。Respectable Software 使用這個簽署金鑰來簽署 MIDlet 套件，並將憑證隨 MIDlet 套件一起發行。當 MIDlet 套件安裝在模擬器或裝置上時，實作會使用它自己的 Super-Trustee 根憑證來檢驗 Respectable 的憑證。然後使用 Respectable 的憑證檢驗 MIDlet 套件上的簽名。假定所有項目都通過檢查，裝置或模擬器就會將 MIDlet 套件安裝到與 Super-Trustee 根憑證相關的任何保護網域，很可能為 identified\_third\_party。

工具組提供一些工具，可簽署 MIDlet 套件、管理金鑰及管理根憑證。

### 6.2.1 MSA 保護網域

工具組支援五個 MSA 的保護網域：

- unidentified\_third\_party - 為無法判定來源與可靠性的應用程式提供高階安全性。當應用程式嘗試機密性操作時，會經常提示使用者
- identified\_third\_party - 適用於使用加密憑證判斷其來源的 MIDlet。權限不會自動授予，但是使用者收到的提示比在 unidentified\_third\_party 網域中時少。
- manufacturer - 適用於其憑證來自製造商的根憑證的 MIDlet 套件。
- minimum - 此網域中的 MIDlet 不得獲得任何權限。
- maximum - 此網域中的 MIDlet 可獲得所有權限。

當您按 [ 執行 ] 按鈕，在模擬器中執行應用程式時，程式碼會預設在 unidentified\_third\_party 保護網域中執行。

## 6.2.2 Java for the Wireless Toolkit Industry 保護網域

Sun Java Wireless Toolkit 包含四個保護網域：

- `untrusted` - 為無法判定來源與可靠性的應用程式提供高階安全性。當應用程式嘗試機密性操作時，會經常提示使用者。
- `trusted` - 此網域中的 MIDlet 可獲得所有權限。
- `minimum` - 此網域中的 MIDlet 不得獲得任何權限。
- `maximum` - 此網域中的 MIDlet 可獲得所有權限 (相當於 `trusted`)。

當您按 [ 執行 ] 按鈕，在模擬器中執行應用程式時，程式碼會預設在 `untrusted` 保護網域中執行。

## 6.3 簽署 MIDlet 套件

若要簽署 MIDlet 套件，您必須先封裝該套件 (選取 [ 專案 ] > [ 封裝 ])，然後選擇 [ 專案 ] > [ 簽署 ]。就會顯示簽署視窗 (圖 6-2)。

圖 6-2 MIDlet 套件簽署視窗



只要在 [ 別名清單 ] 中選取要使用的金鑰，再按一下 [ 簽署 MIDlet 套件 ] 按鈕 (綠色三角形) 即可。

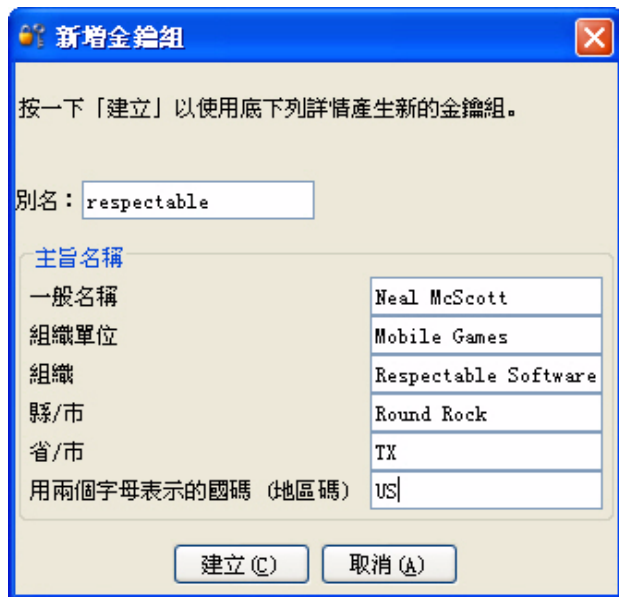
## 6.4 管理金鑰

MIDlet 簽署視窗也可以用來管理金鑰。爲了測試，您可以建立金鑰組，以簽署 MIDlet。若要在裝置上部署，必須向裝置能辨識的憑證授權單位取得簽署金鑰組。您也可以從既有的 Java SE 平台金鑰庫匯入金鑰。

### 6.4.1 建立新的金鑰組

若要建立全新的金鑰組，請按一下 [ 新增金鑰組 ]。工具組會提示您輸入金鑰別名及金鑰組的相關資訊。

圖 6-3 建立新的金鑰組



新增金鑰組

按一下「建立」以使用底下列詳情產生新的金鑰組。

別名:

主旨名稱

一般名稱	<input type="text" value="Neal McScott"/>
組織單位	<input type="text" value="Mobile Games"/>
組織	<input type="text" value="Respectable Software"/>
縣/市	<input type="text" value="Round Rock"/>
省/市	<input type="text" value="TX"/>
用兩個字母表示的國碼 (地區碼)	<input type="text" value="US"/>

按一下 [ 建立 ] 之後，工具組會提示您選擇保護網域。您剛剛建立的金鑰組與保護網域之間的連線可能有些迂迴，但有充分的道理：

- 工具組使用剛剛建立的金鑰組建立自我簽署的根憑證。
- 根憑證加入模擬器的根憑證清單中。
- 工具組必須將根憑證與保護網域建立關聯。

現在試想，如果安裝使用新金鑰簽署的 MIDlet 套件，會發生什麼狀況：

- 實作會檢查 MIDlet 套件描述元中的憑證鏈。在本案例中，憑證鏈是單一憑證，亦即自我簽署的根憑證。
- 實作嘗試在其內部清單中尋找憑證鏈的根。此動作會成功，因為當您建立金鑰組時已加入根憑證。
- 實作認為憑證有效，並使用它來檢驗 MIDlet 套件上的簽名。
- MIDlet 套件安裝在您挑選的任何保護網域中。

## 6.4.2 獲取真實金鑰

在 Sun Java™ Wireless Toolkit for CLDC 環境中建立金鑰組和簽署 MIDlet 的能力，僅做為測試用途。當您在實際裝置上執行應用程式時，必須向裝置能辨識的憑證授權單位取得簽署金鑰組。

以真實金鑰簽署 MIDlet 套件的程序運作方式如下：

### 1. 產生新的金鑰組。

在 Sun Java™ Wireless Toolkit for CLDC 中，可藉由在 MIDlet 簽署視窗中按下 [ 新增金鑰組 ] 達成，如上所述。

### 2. 產生一個憑證簽署請求 (Certificate Signing Request, CSR)。

a. 在簽署視窗中按下 [ 產生 CSR ]。

b. 要改變 CSR 檔案的位置，輸入一個新的路徑或按下 [ 瀏覽 ] 並選擇一個新的檔案位置。

c. 按下 [ 建立 ] 以寫入 CSR 檔。

在 CSR 檔案寫完後，會顯示表示成功寫入的訊息。

### 3. 將 CSR 傳送到憑證授權單位 (CA)。

CA 需要您提供更多資訊以驗證您的身份。您也必須支付 CA 產生憑證的費用。

CA 確認您的身份並收到費用後，會傳送可驗證公開金鑰的憑證。

### 4. 在 MIDlet 簽署視窗中按下 [ 匯入憑證 ... ]，將憑證匯入 Sun Java™ Wireless Toolkit for CLDC。

現在您就可以使用您的私密金鑰簽署 MIDlet 套件了。Sun Java™ Wireless Toolkit for CLDC 將會處理將簽署及您的憑證存放至 MIDlet 套件的細節。

### 6.4.3 匯入現有的金鑰組

您可能想要將 Java SE 平台金鑰庫中的金鑰用於 MIDlet 簽署。在這種狀況下，您必須將簽署金鑰匯入 Sun Java™ Wireless Toolkit for CLDC，讓您可以簽署您的 MIDlet 套件。若要從 MIDlet 簽署視窗執行此動作，請按一下 [匯入金鑰組]。選取包含 Java SE 平台金鑰庫的檔案。系統會提示您選取要匯入的金鑰組之別名，然後您僅需在此金鑰組匯入金鑰庫後，輸入金鑰組的別名以便識別即可。最後，您必須選取金鑰組根憑證的保護網域。

### 6.4.4 移除金鑰組

若要從 MIDlet 簽署視窗中移除金鑰組，請選取別名，並選擇 [動作] > [刪除選項]。

---

## 6.5 管理憑證

這一節描述如何使用 Sun Java™ Wireless Toolkit for CLDC，管理模擬器的根憑證清單。

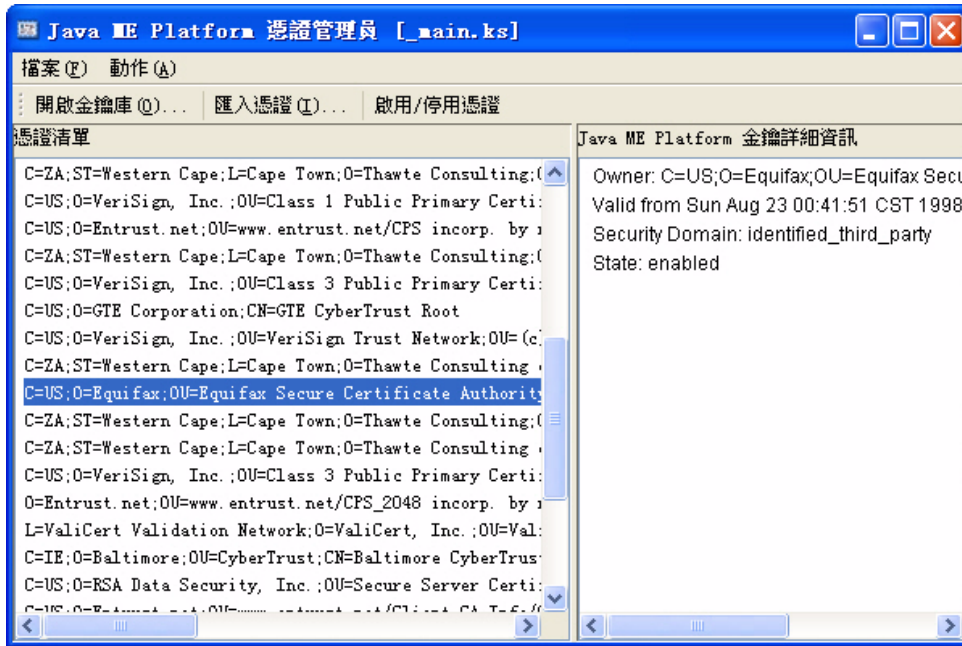
實際裝置具有類似的根憑證清單，但是使用者無法修改。當您想在實際裝置上部署應用程式時，您必須使用裝置上存在之根憑證的憑證授權單位所發出的簽署金鑰。否則，裝置將無法檢驗您的應用程式。

當您在部署應用程式時，工具組的憑證管理公用程式會提供便利的方式，讓您測試處理模擬器的根憑證清單。

選擇 [檔案] > [公用程式]。選取 [管理憑證]，並按下 [啟動] 以開啓憑證管理員視窗。會顯示微金鑰庫 `_main.mks`。金鑰庫位於 `appdb` 目錄中。

`appdb` 目錄也包含 `keystore.ks` 和 `serverkeystore.ks`。Java ME 平台憑證管理員無法開啓 `*.ks` 檔案，但是如第 6-8 頁的第 6.5.2 節「匯入憑證」所述，您可以從這些金鑰庫匯入憑證。

圖 6-4 憑證管理員



視窗左側【憑證清單】的每一行分別顯示一個憑證。當您按一下憑證時，其詳細資訊會顯示在視窗右側，還會顯示與憑證相關聯的保護網域。

## 6.5.1 啟用和停用憑證

憑證可以啟用也可以停用。如果您想讓憑證暫時不可用卻又不想從金鑰庫移除，這是很方便的功能。若要啟用或停用憑證，請在清單中選取它，並按下【啟用 / 停用憑證】。工具組會要求您確認動作。選擇【是】以繼續。

## 6.5.2 匯入憑證

您可以從憑證檔案或從 Java SE 平台金鑰庫檔案匯入憑證。

若要從檔案匯入憑證，請在憑證管理員視窗中按一下【匯入憑證】。找到憑證檔案後，選擇與憑證相關的保護網域。

若要從 Java SE 平台金鑰庫匯入憑證，請在憑證管理員視窗中選擇【動作】>【匯入 Java SE 憑證】。首先，選擇憑證的保護網域。然後選取金鑰庫檔案，並輸入金鑰庫密碼。最後，選取要匯入的憑證別名。



### 6.5.3 移除憑證

若要從清單中移除憑證，請選取憑證，選擇 [ 動作 ] > [ 刪除選項 ]。

---

## 6.6 支援 USB 加密碟

USB 加密碟為公開與私密金鑰和憑證提供可攜帶且受密碼保護的儲存方式。Java SE PKCS#11 原生介面可存取的 USB 加密碟，需具有與 PKCS#11 相容的原生驅動程式。安裝此驅動程式後，PKCS#11 程式庫即包含於其中。在 Windows 中，此程式庫為 win32 DLL。

本小節提供在 Windows 平台中安裝和使用 USB 加密碟的範例說明。

---

**備註** – 不支援 Linux，因為尚未使用 Linux 驅動程式對 USB 加密碟進行全面測試。如果有與 PKCS#11 相容的原生驅動程式，則 USB 加密碟也許能用在 Linux 中。

---

本小節其餘部分逐步介紹 USB 加密碟支援所需的安裝和設定程序。

### 6.6.1 安裝 USB 加密碟驅動程式

關閉所有應用程式。

1. 請至 <http://downloads.geotrust.com/TCSPIKEY0407203016.exe>
2. [ 檔案下載 ] 對話方塊開啟時，按一下 [ 儲存 ]。  
記下儲存該可執行檔的目錄。
3. 選取該可執行檔，然後連按兩下該可執行檔案開始 **Crypto Token** 安裝。  
遵循安裝提示。插入加密碟以便完成安裝。
4. **Windows** 新增硬體精靈啟動。  
遵循指示接受所有預設動作。
5. 精靈結束後，選取 [ 是 ] 以重新啟動電腦。

## 重設 USB 加密碟密碼

此步驟僅對新 USB 加密碟有效。所有 USB 加密碟均隨附有預設的密碼：PASSWORD (全大寫)。您應重設此密碼。

1. 在 **Windows** 工作列上按一下 [開始]。選取 [所有程式] > [GeoTrust Token] > [iKey 2000 Series Software] > [PassPhrase Utility]。
2. 按一下 [更新密碼]。  
在輸入新密碼前，系統會提示您輸入「舊密碼」。
3. 重設密碼。  
GeoTrust 和 Cingular 建議您將密碼設為至少八個混合字元的組合。

## 管理 USB 加密碟

請至 GeoTrust 驅動程式安裝目錄。執行 CIPUtils.exe 以管理 USB 加密碟的內容。

### 6.6.2 使用 USB 加密碟

[簽署 MIDlet 套件] 對話方塊提供了 USB 加密碟的存取權。連接 USB 加密碟並安裝了驅動程式後，請選取 [檔案] > [載入金鑰庫] > [來自 USB 加密碟] (或鍵入 Ctrl-T)。如果 USB 加密碟是受密碼保護的，則系統會提示您輸入一個密碼。正確載入加密碟後，即會列出所有的別名和金鑰詳細資訊。然後，您便可如第 6-4 頁的第 6.3 節「簽署 MIDlet 套件」所述選取金鑰並使用它們進行簽署。

嘗試從 USB 加密碟載入金鑰庫時可能會看到錯誤訊息。

- 如果無法存取該 USB 加密碟，您可能會看到錯誤「USB token or driver might be unplugged or invalid (未插入 USB 加密碟或未安裝驅動程式，或驅動程式無效)」。確保已插入磁碟。您可能需要嘗試機器上其他的 USB 連接埠。
- 如果未找到原生程式庫，系統會提示您輸入隨驅動程式一起安裝的 DLL 程式庫的路徑。通常位於驅動程式的安裝目錄。

## 使用 Wireless Messaging API

---

Sun Java™ Wireless Toolkit for CLDC 以一個先進的模擬環境，支援 Wireless Messaging API (WMA)。WMA 1.1 (JSR 120) 讓 MIDlet 可以傳送與接收簡訊服務 (Short Message Service, SMS) 或網內廣播服務 (Cell Broadcast Service, CBS) 訊息。WMA 2.0 (JSR 205) 也包含 MMS 訊息的支援。

這一章描述可用來開發 WMA 應用程式的工具。一開始，先示範如何配置模擬器的 WMA 支援。接著，會描述 WMA 主控台，這是測試 WMA 應用程式的一個便利公用程式。本章最後一節簡要描述網路監視器的 WMA 支援。

---

### 7.1 設定模擬器的電話號碼

模擬器的每個執行中實例都有一個模擬的電話號碼，顯示在模擬器視窗的標題列。電話號碼很重要，因為這些號碼要做為 WMA 訊息的位址。依預設，第一個模擬器實例的電話號碼是 +555000；後續的模擬器實例會依遞增順序使用唯一的號碼：+5550001、+5550002、+5550003 等。

若要改變指定的電話號碼，請選擇 [ 編輯 ] > [ 偏好 ] 並在左側窗格中選取 [WMA]。

圖 7-1 設定 WMA 偏好



【下一個模擬器的電話號碼】欄位的名稱就已經清楚說明其涵義。如果在此欄位中填入號碼，下一個模擬器實例就會使用該號碼。

如果【下一個模擬器的電話號碼】已經使用中，或欄位空白，則下一個模擬器實例就會使用【優先指定的電話號碼】。後續的實例依序遞增。

例如，假設在【下一個模擬器的電話號碼】中輸入 +6269333，在【優先指定的電話號碼】中輸入 +5550000。如果您啟動四個模擬器實例，則其號碼為 +6269333、+5550000、+5550001 和 +5550002。

---

## 7.2 模擬不可靠的網路

傳送長訊息時，必須先將訊息分段，分別傳送段落之後，在接收端重新組合起來。您可以在 Sun Java™ Wireless Toolkit for CLDC 中模擬無線網路的一些嚴重問題。同上，請選擇 [ 編輯 ] > [ 偏好 ] 並選取 [WMA]。

如果希望工具組丟掉一些訊息段落，請將 [ 隨機訊息分割遺失 ] 滑動軸調整成需要的百分比。如果要模擬訊息段落傳送與接收時間之間的延遲，請以毫秒為單位，在 [ 訊息分割傳送延遲 ] 欄位中輸入延遲。

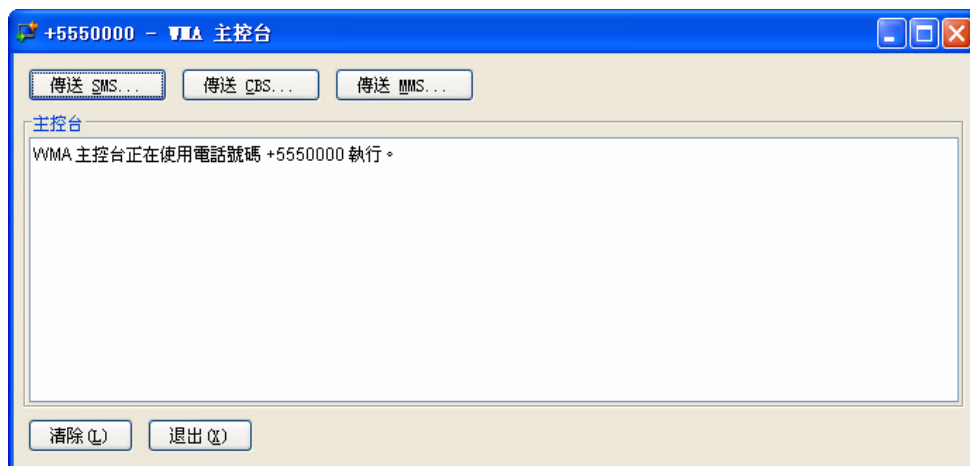
---

## 7.3 使用 WMA 主控台傳送訊息

WMA 主控台是便利的公用程式，可讓您傳送與接收訊息。例如，您可以用 WMA 主控台傳送 SMS 訊息，給在模擬器上執行的 MIDlet。

若要啟動 WMA 主控台，請選擇 [ 檔案 ] > [ 公用程式 ]。在 WMA 主控台上按一下，並按下 [ 啟動 ]。

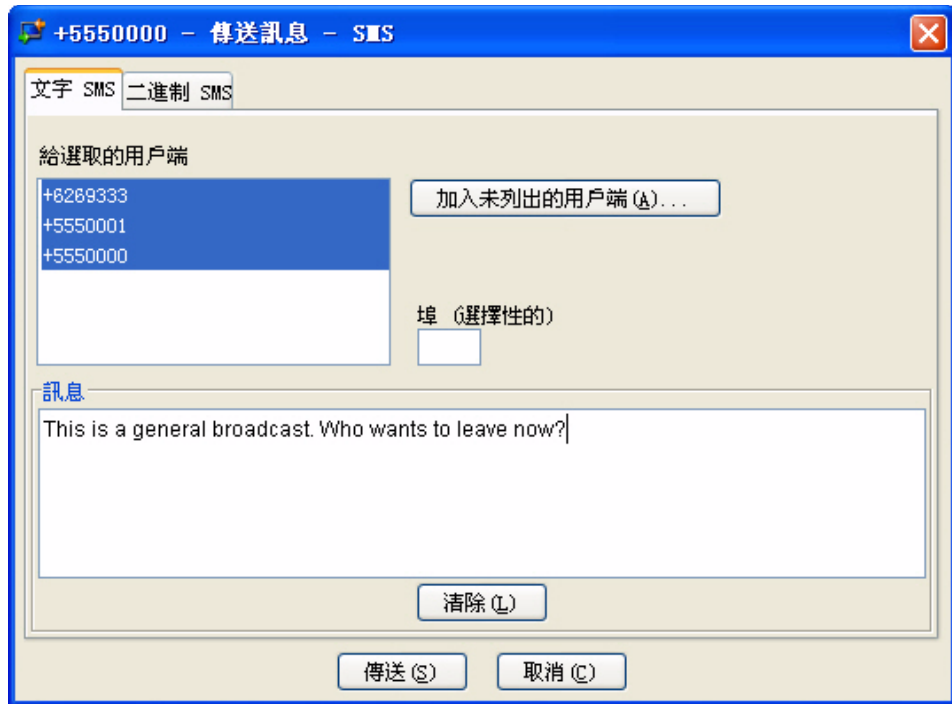
圖 7-2 WMA 主控台



## 7.3.1 傳送文字的 SMS 訊息

若要傳送文字的 SMS 訊息，請按一下 [ 傳送 SMS ]。會顯示傳送視窗。

圖 7-3 傳送文字訊息

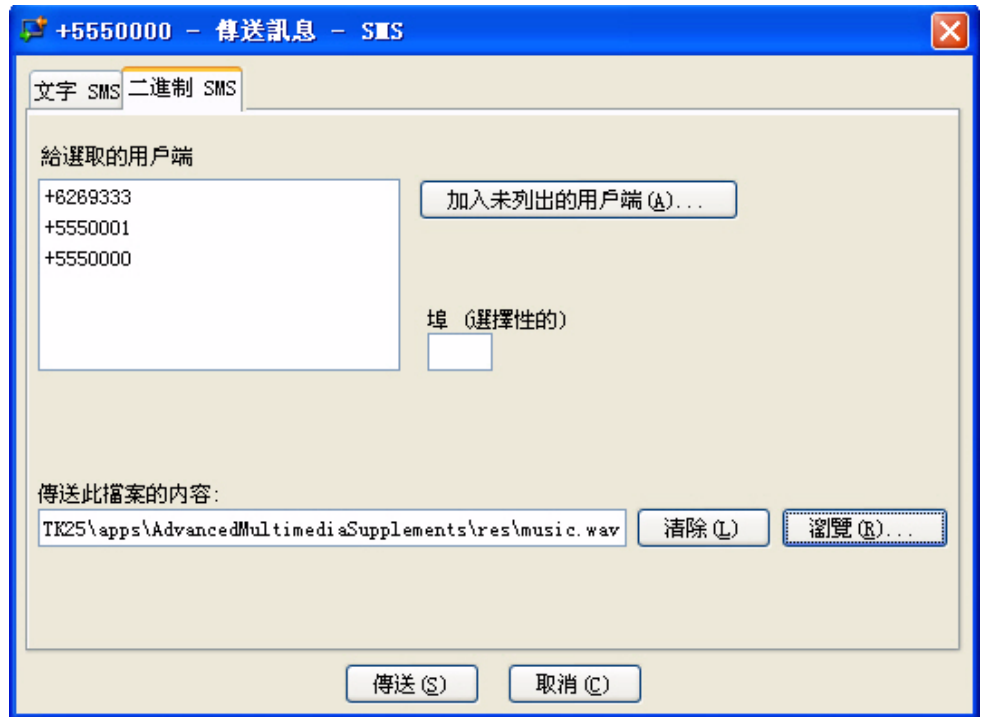


視窗自動列出所有執行中模擬器實例的電話號碼。選取目標 (按一下時如按住 Ctrl 可選取多個目標)，亦可輸入連接埠號。輸入訊息，並按一下 [ 傳送 ]。

## 7.3.2 傳送二進位 SMS 訊息

您可以使用 WMA 主控台以二進位訊息的方式傳送檔案內容。按一下 [傳送 SMS] 開啟傳送視窗。按一下 [二進位 SMS] 標籤。

圖 7-4 傳送二進位訊息

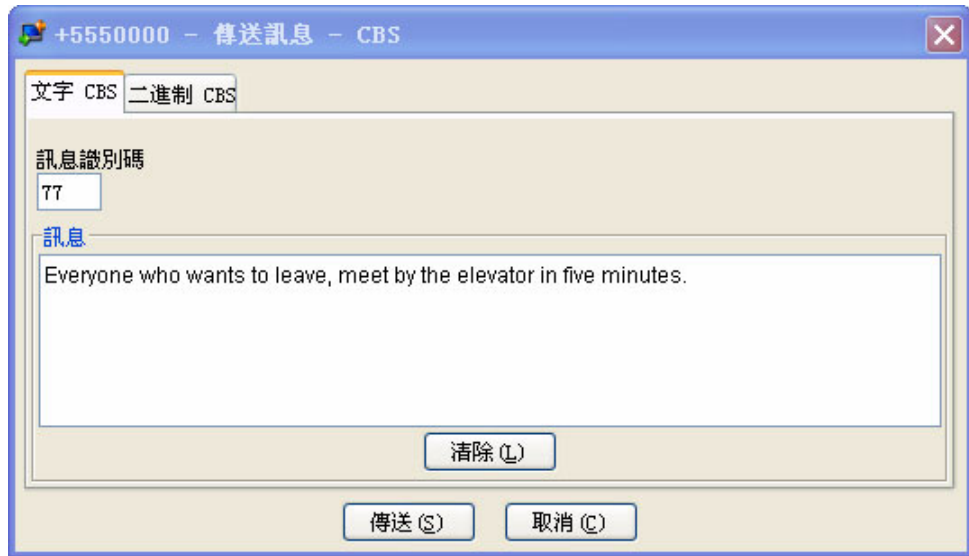


選取收件者的方式與傳送文字的 SMS 訊息時相同。您可以直接輸入檔案路徑，或按一下 [瀏覽] 開啟檔案選擇器。

### 7.3.3 傳送文字或二進位 CBS 訊息

傳送 CBS 訊息與傳送 SMS 訊息的方式類似，只不過您不必選擇收件者。若要傳送文字或二進位 CBS 訊息，請在 WMA 主控台中按一下 [ 傳送 CBS ]。就會顯示傳送視窗。

圖 7-5 傳送 CBS 訊息



### 7.3.4 傳送 MMS 訊息

MMS 訊息包含一個或多個檔案，通常是影像或聲音。MMS 訊息可傳送給多個收件者。若要從 WMA 主控台傳送 MMS 訊息，請按一下 [ 傳送 MMS ] 按鈕。

撰寫 MMS 訊息的視窗有兩個標籤，一個用於收件者，一個用於內容。一開始要先填入主旨與收件者。如果要新增其他收件者，請按一下 [ 增加 ] 按鈕。例如，要傳送訊息給號碼為 +5550001 的執行中模擬器，就要在 [ 收件者 ] 行填入 `mms://+5550001`。若要移除收件者，請先選取該行，再按一下 [ 移除 ]。



圖 7-6 新增 MMS 訊息的收件者

**+5550000 - 傳送訊息 - MMS**

標頭 部分

主旨: Group Meeting Tuesday 10 a.m.

應用程式 ID:

收件者	mms://5550000
收件者	mms://6269333

添加 (A) 移除 (R)

傳送 (S) 取消 (C)

若要在訊息中新增媒體檔，請按一下【部分】標籤。按一下【增加】為訊息新增其他部分。若要移除部分，請選取後再按下【移除】。

圖 7-7 在 MMS 訊息中新增部分

**+5550000 - 傳送訊息 - MMS**

標頭 部分

檔案	內容位置	內容 ID	Mime 類型	編碼	大小
C:\WTK25\...\logo.png	logo.png	0	image/png	MS950	1692

總大小: 1692

添加 (A) 移除 (R)

傳送 (S) 取消 (C)

## 7.4 在 WMA 主控台中接收訊息

WMA 主控台也可以接收訊息。WMA 主控台視窗在標題列上有它自己的電話號碼。您可以從模擬器上執行的應用程式傳送訊息給 WMA 主控台。

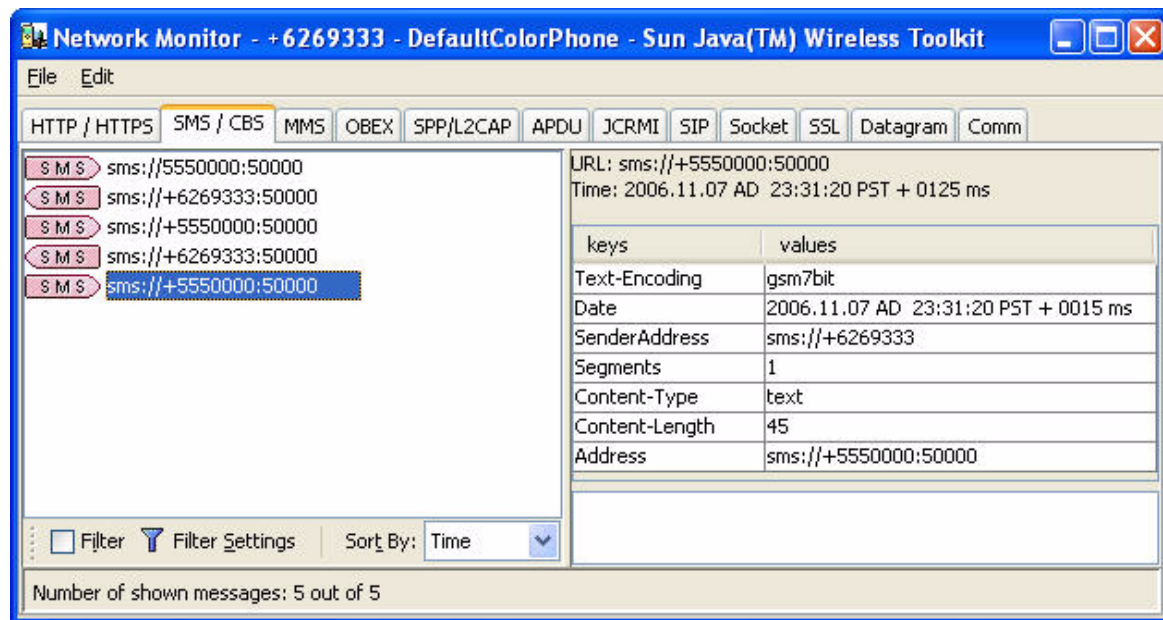
接收的訊息顯示在 WMA 主控台的文字區域。

## 7.5 與 WMA 一起使用網路監視器

網路模擬器在 [第 5 章](#) 中有詳盡描述。您可以使用網路監視器追蹤傳入 / 傳出模擬器的 WMA 訊息。

按一下 [SMS/CBS] 或 [MMS] 標籤就可以看到 WMA 訊息。網路監視器的左側窗格中顯示訊息及其段落的相關資訊。按一下訊息或訊息段落，可在右側窗格中查看其詳細資訊。

圖 7-8 使用網路監視器檢視 WMA 訊息



## 使用 Mobile Media API

JSR 135 (Mobile Media API (MMAPI)) 提供標準的 API，供您描繪與擷取時間型式的媒體，例如音效或視訊。此 API 的設計可彈性適應不同裝置支援的媒體格式、協定與功能。如需有關使用 MMAPI 進行程式設計的資訊，請參閱以下文章：

■ Mobile Media API Overview

[http://developers.sun.com/techtopics/mobility/apis/articles/mmapi\\_overview/](http://developers.sun.com/techtopics/mobility/apis/articles/mmapi_overview/)

■ The J2ME Mobile Media API

<http://developers.sun.com/techtopics/mobility/midp/articles/mmapioverview/>

### 8.1 支援的格式與協定

模擬器的 MMAPI 實作支援下列媒體類型。

表格 8-1 支援的 MMAPI 媒體格式

MIME 類型	描述
audio/amr	Adaptive Multi-Rate
audio/midi	MIDI 檔案
audio/sp-midi	Scalable Polyphony MIDI
audio/x-tone-seq	MIDP 2.0 鈴聲序列
audio/x-wav	WAV PCM 取樣音效
image/gif	GIF 89a (GIF 動畫)
video/mpeg	MPEG 視訊
video/vnd.sun.rgb565	視訊擷取

---

## 8.2 Adaptive Multi-Rate (AMR) 內容

Sun Java™ Wireless Toolkit for CLDC 模擬對 Adaptive Multi-Rate (AMR) 內容 (<http://www.ietf.org/rfc/rfc3267.txt>) 的支援。儘管工具組無法解碼 AMR 內容，但實作將在收到請求時為 AMR 內容傳回播放程式。

### 8.2.1 Windows

在 Windows 上，AMR 檔案將轉換為一般 WAVE 檔案並傳送至 Qsound。由於 Windows 版本與 3GPP 實作可互通，因此您無須執行任何動作即可播放 AMR 檔案。

### 8.2.2 Linux

Linux 的 AMR 支援以 3GPP AMR Narrow Band (AMR-NB) Reference Implementation 解碼器與 SOX 音效處理器 (不支援 AMR Wide Band) 為基礎。

#### 8.2.2.1 啟用 AMR 支援

遵循以下步驟啟用 AMR 支援。

1. 獲得 3GPP 提供的 AMR-NB RI。

有若干可用的版本。其中的一個版本位於：

[http://www.3gpp.org/ftp/Specs/archive/26\\_series/26.073/26073-530.zip](http://www.3gpp.org/ftp/Specs/archive/26_series/26.073/26073-530.zip)

2. 開啟封裝中包含的 **makefile**。

找到以「CFLAGS =」開頭的行。

- a. 增加 DMMS\_IO 選項。

- b. 移除 pedantic-errors 選項。

儲存與編譯。如果出現問題，另請嘗試移除 O3 (或 O2) 最佳化旗標 (二進位將近乎大小的兩倍)。

3. 若要生成 RI，請輸入：`make VAD=VAD1`

編譯後，您應具有一個名稱為 `decoder` 的二進位檔案。其為 AMR-NB 解碼器本身。

4. 設定環境變數 `AMR_DECODER` 以指向解碼器路徑。  
例如，如果 `~/amr` 是該解碼器的路徑，請指定：  

```
export AMR_DECODER=~/amr/decoder
```
5. 設定 `decoder` 檔案的執行存取權限。  
在該解碼器的同一目錄下，鍵入：  

```
chmod 555 ./decoder
```
6. 按以下步驟測試解碼器：  
如第 A-28 頁的第 A.13 節「[MobileMediaAPI](#)」所述，執行 `MobileMediaAPI` 示範 `Simple Player`。從主功能表中選取 `[Simple Player]`，然後選取「`AMR Narrow Band [jar]`」。請記得打開喇叭。

### 8.2.2.2 AMR 格式支援

某些 AMR-NB 的「複製」版本也可能有問題。AMR-NB 編碼解碼器資料似乎至少有兩種檔案格式：

- 儲存為「AMR 檔案儲存格式」的 `.AMR` 檔案。  
其指定於 `draft-ietf-avt-rtp-amr-10.txt` 第 6.2 節，涵蓋在 Ericsson AMR 轉換器工具中。(此草案為 RFC 3267 的早期形式)。這些檔案由 Ericsson AMR 工具與 Nokia Series 60 電話處理。它們的標頭為 `#!AMR\n`，並使用 `big-endian` 編碼。
- 儲存為「AMR 介面格式 2」的 `.COD` 檔案  
此格式指定於附錄 A 的 3GPP TS 26.101。這些檔案可由 3GPP TS 26.104 浮點參考編碼解碼器來源套裝軟體編碼與解碼。這些檔案沒有標頭，並使用 `little-endian` 編碼。

若要在這些格式間相互轉換，請嘗試此 Python 程序檔：

<http://www.connectivity.com/~eaw/amrwork/amrconv.py>

---

## 8.3 使用 MediaControlSkin

Sun Java™ Wireless Toolkit for CLDC 隨附一個著重於多媒體播放與控制的模擬器面板 `MediaControlSkin`。此面板包括的按鈕上，有代表播放、停止、音量大小和其他指令的符號。若要瞭解 `MediaControlSkin` 的用處，請以 `MobileMediaAPI` 示範應用程式試用。

---

## 8.4 媒體擷取

Sun Java™ Wireless Toolkit for CLDC 模擬器支援音效與視訊擷取。音效擷取的支援方式，是利用模擬器執行系統的擷取功能。

視訊擷取的支援方式是模擬相機輸入。

如需示範如何擷取音效與視訊的詳細資訊與原始碼，請參考 MobileMediaAPI 應用程式範例。

---

## 8.5 正常 MIDlet

MIDlet 有 MIDP 規格所定義的生命週期。MIDlet 可因事件而暫停，如來電事件。正常 MIDlet 會在暫停時釋出重要裝置資源，並在 MIDlet 重新繼續時重新配置或重新啟動這些資源。在 MMAPI 中，任何描繪內容的〔播放程式〕應在 MIDlet 暫停時停止。

Sun Java™ Wireless Toolkit for CLDC 會在您暫停 MIDlet 但其執行中的〔播放程式〕卻沒有停止時，於主控台上顯示訊息。您可使用在 MobileMediaAPI 示範應用程式中的 Pausing Audio Test MIDlet 來測試這項功能。如需詳細資訊，請參閱[附錄 A](#)。

每一執行中模擬器的警告訊息僅顯示一次。

---

## 8.6 鈴聲

如第 A-29 頁的第 A.13.1 節「Simple Tones」與第 A-29 頁的第 A.13.2 節「Simple Player」所示，MMAPI 可用於播放鈴聲。常用的鈴聲格式有好幾種。您可以下載鈴聲或建立自己的鈴聲。

### 8.6.1 下載鈴聲

鈴聲可從許多網站下載，包括：

- <http://www.surgeryofsound.co.uk/>
- <http://www.convertyourtone.com/>
- <http://www.filmfind.tv/ringtones/>

## 8.6.2 鈴聲格式

本小節提供了若干格式範例

- RTTTL ( 鈴聲文字傳輸語言格式 ) 的解釋位於：

<http://www.convertyourtone.com/rtttl.html>

- Nokia Composer

以下提供了 Nokia Composer 格式的貝多芬第九號交響曲：

16g1,16g1,16g1,4#d1,16f1,16f1,16f1,4d1,16g1,16g1,16g1,16#d1,  
16#g1,16#g1,16#g1,16g1,16#d2,16#d2,16#d2,4c2,16g1,16g1,16g1,  
16d1,16#g1,16#g1,16#g1, 16g1,16f2,16f2,16f2,4d2

- Ericsson Composer

貝多芬 G 大調小步舞曲：

a b + c b + c b + c b + C p + d a B p + c g A  
p f g a g a g a g A p b f G p a e F

貝多芬第九號交響曲主旋律：

f f f # C # d # d # d C p f f f # c # f #f # f f + # c + #  
c + # c # A ff f c # f # f # f f + # d + # d + # d

- Siemens Composer 格式

Inspector Gadget 主旋律：

C2(1/8) D2(1/16) Dis2(1/8) F2(1/16) G2(1/8)  
P(1/16) Dis2(1/8) P(1/16) Fis2(1/8) P(1/16)  
D2(1/8) P(1/16) F2(1/8) P(1/16) Dis2(1/8)  
P(1/16) C2(1/8) D2(1/16) Dis2(1/8) F2(1/16)  
G2(1/8) P(1/16) C3(1/8) P(1/16) B2(1/2) P(1/4)  
C2(1/8) D2(1/16) Dis2(1/8) F2(1/16) G2(1/8) P(1/16)  
Dis2(1/8) P(1/16) Fis2(1/8) P(1/16) D2(1/8) P(1/16)  
F2(1/8) P(1/16) Dis2(1/8) P(1/16) C3(1/8) B2(1/16)  
Ais2(1/8) A2(1/16) Gis2(1/2) G2(1/8) P(1/16) C3(1/2)

- Motorola Composer

貝多芬第九號交響曲：

4 F2 F2 F2 C#4 D#2 D#2 D#2 C4 R2 F2 F2 F2 C#2 F#2 F#2  
F#2 F2 C#+2 C#+2 C#+2 A#4 F2 F2 F2 C2 F#2 F#2 F#2 F2  
D#+2 D#+2 D#+2

## ■ Panasonic Composer

貝多芬第九號交響曲：

```
444** 444** 444** 1111* 4444** 4444** 4444** 111*  
0** 444** 444** 444** 1111** 4444** 4444** 4444**  
444** 11** 11** 11** 6666* 444** 444** 444** 111**  
4444** 4444** 4444** 444** 22** 22** 22**
```

## ■ Sony Composer

貝多芬第九號交響曲：

```
444****444****444****111#*****444#*****444#*****444#*****  
111*****(JD) 0000444****444****444****111#*****444#*****  
444#*****444#*****444****11#*****11#*****11#*****666#*****  
444****444****444****111****444#*****444#*****  
444#*****444****22#*****22#*****22#*****
```



## 使用 Mobile Graphics

---

這一章簡介如何使用圖形內容。Sun Java™ Wireless Toolkit for CLDC 提供三個 API，以提供完整的互動 2D 與 3D 圖形能力：

- Mobile 3D Graphics API for J2ME (JSR 184) 使用低階層 API 與高階層場景圖形 API 提供 3D 圖形能力。這一章將提供使用 JSR 184 的簡介與一般指南。
- Scalable 2D Vector Graphics API for J2ME (JSR 226) 可描繪精緻互動式的 2D 內容。
- Java Bindings for OpenGL® ES (JSR 239) 可向開放式標準 OpenGL® ES 圖形 API 提供 Java 語言介面。

---

### 9.1 使用 Mobile 3D Graphics API

JSR 184 是定義 Mobile 3D Graphics (M3G) API for the J2ME 的規格。這個 API 以精簡的套裝軟體，提供適合 CLDC/MIDP 裝置的 3D 功能。此 API 提供二種顯示 3D 圖形內容的方法。立即模式 API 讓應用程式能夠立即建立與處理 3D 元素。在這一層之上是場景圖形 API (又稱為保留模式)，它可讓應用程式能夠載入與顯示事先設計好的完整 3D 場景。應用程式可以自由使用最適合的方法，或混合使用立即模式與保留模式 API。JSR 184 規格也為場景圖形定義一種檔案格式 (.m3g)。

如需詳細資訊，請於

<http://jcp.org/en/jsr/detail?id=184> 參閱 JSR 184 規格。

JSR 184 為 CLDC/MIDP 裝置提供標準的 API，促成了新一代的 3D 應用程式。接著立即模式與 OpenGL ES (3D 圖形的標準輕型 API) 相容。如需 OpenGL ES 的詳細資訊，請參閱 <http://khronos.org/>。

## 9.1.1 立即模式

立即模式適合用於以演算法產生 3D 圖形內容的應用程式，例如科學可視化或統計圖形。應用程式會建立 3D 物件，並直接進行處理。

如需立即模式的範例，請參閱 Demo3D 應用程式範例中的 Life3D MIDlet。

## 9.1.2 保留模式

大部分應用程式（尤其是遊戲）使用保留模式或稱場景圖形 API。在此方法中，圖形設計者或藝術家使用 3D 模型建立軟體，來建立場景圖形。場景圖形儲存成 JSR 184 格式的檔案。場景圖形檔與應用程式隨附在一起。在執行階段，應用程式使用場景圖形 API 載入及顯示檔案。

應用程式可以處理載入場景圖形的不同部分，讓角色更生動或建立其他效果。基本策略是儘可能在模型建立軟體中處理各項工作。在執行階段，應用程式可以抓取並處理場景圖形的不同部分，這也包括動畫或其他效果的路徑。

如需保留模式的範例，請參閱 Demo3D 應用程式範例中的 retainedmode MIDlet。

## 9.1.3 品質與速度之間的取捨

開發 MIDP 時的一大難題，是一般裝置的環境都受到限制。與桌上型電腦相比，MIDP 裝置的處理器速度較慢，記憶體也比較少。這些難題也一樣影響 3D 圖形。為配合各式各樣不同的實作，JSR 184 規格提供各種機制，以儘可能提高 3D 場景的顯示效率。

一種作法是 *scoping*（範圍），這種技術的作法是由您告訴 3D 圖形實作，物件何時不會彼此互動。例如，如果定義一個房子的場景圖形，就可以用 *scoping* 指定地下室的光線不會影響二樓臥室的外觀。*Scoping* 會減少顯示某個場景所需的計算次數，讓實作的工作變得比較容易。

但是一般而言，提高 3D 場景描繪速度的最佳方法，是犧牲一些品質。Mobile 3D Graphics API 包含描繪提示，讓應用程式可以建議實作如何犧牲品質，以改善描繪速度。

## 9.1.4 建立 Mobile 3D Graphics 內容

大部分行動 3D 應用程式會使用資源檔案中的場景圖形，來描述物件、場景與角色。場景圖形的建立，通常是由圖形設計者或藝術家（而不是程式設計者）使用標準 3D 模型工具來進行。

有數家供應商提供工具，可編寫內容，並將檔案轉換成 JSR 184 格式。Superscape (<http://superscape.com/>) 就是供應商之一。

因為在使用立即模式 API 的應用程式中，不太容易建立與處理 3D 圖形內容，所以大部分應用程式會儘可能依靠場景圖形檔。藉由儘可能在設計時將最多的內容放入場景圖形檔，將會大量簡化應用程式在執行階段的工作。

---

## 9.2 描繪可縮放式向量圖形內容

可縮放式向量圖形 (SVG) 是由全球資訊網協會定義的標準。它是用來描繪精緻互動式 2D 圖形的 XML 語法。

Sun Java™ Wireless Toolkit for CLDC 模擬器支援 JSR 226，即 Scalable 2D Vector Graphics API for J2ME。JSR 226 是用於載入、處理、描繪與播放 SVG 內容的 Java ME API。SVG Tiny 使用精簡但功能強大的 XML 格式，描繪精緻互動式的動畫 2D 內容。

儘管使用文字編輯器也可製作 SVG 內容，但大多數使用者更喜歡使用編寫工具。以下提供了三種工具：

- **BeatWare Mobile Designer** -  
[http://www.beatware.com/products/md\\_golive.html](http://www.beatware.com/products/md_golive.html)
- **Ikivo Animator** - <http://www.ikivo.com/animator/>
- **Adobe Illustrator CS2** -  
<http://www.adobe.com/products/illustrator/main.html>

使用 SVG 內容的 Java ME 應用程式建立的圖形效果，可適應使用者顯示器的顯示解析度和尺寸外型。

SVG 影像可使用兩種方式成為動畫。一種是使用宣告式動畫 (Declarative Animation)。另一種是藉由 API 呼叫重複修改 SVG 影像參數（例如顏色或位置）。[第 A-45 頁的第 A.21.3 節「播放 SVG 動畫」](#)將說明宣告式動畫。

---

## 9.3 OpenGL® ES 簡介

JSR 239 定義兩個 Java 程式設計語言連結：API OpenGL® for Embedded Systems (亦稱為 OpenGL® ES) 與 EGL。OpenGL® ES 屬於 OpenGL®，是用於 3D 圖形的標準 API，普遍應用於桌上型電腦。EGL 是標準平台介面層。OpenGL® ES 與 EGL 均由 Khronos Group (<http://khronos.org/opengles/>) 開發。

物件導向的 JSR 184 要求極強的功能，而 OpenGL® 是低階圖形程式庫，適於存取硬體加速 3D 圖形。請研究 OpenGL®ESDemo 範例專案程式碼。

## 使用 PIM 與 FileConnection API

---

Sun Java™ Wireless Toolkit for CLDC 支援 PDA Optional Packages (PDAP) for the J2ME Platform : JSR 75。JSR 75 包含兩個獨立的 API：

- FileConnection 選擇性套裝軟體讓 MIDlet 能夠存取本機裝置的檔案系統。
- 個人資訊管理 (Personal Information Management, PIM) 選擇性套裝軟體包含用於處理連絡人清單 (通訊錄)、行事曆與待辦清單的 API。

這一章描述 Sun Java™ Wireless Toolkit for CLDC 如何實作 FileConnection 與 PIM API。

---

### 10.1 FileConnection API

在實際裝置上，FileConnection API 讓應用程式可存取儲存在裝置的記憶體或記憶卡上的檔案。

在 Sun Java™ Wireless Toolkit for CLDC 模擬器中，FileConnection API 讓 MIDlet 可存取桌上型電腦硬碟上儲存的檔案。

可用 FileConnection 存取的檔案儲存於 *workdir\appdb\skin\filesystem* 的子目錄中。例如，DefaultColorPhone 模擬器面板隨附一個 root1 根目錄，其中包含一個 Readme 檔案與一個名稱為 photos 的空目錄。該檔案的完整路徑為：

---

**Windows:** *workdir\appdb\skin\filesystem\root1\photos*

**Linux:** *workdir/appdb/skin/filesystem/root1/photos*

---

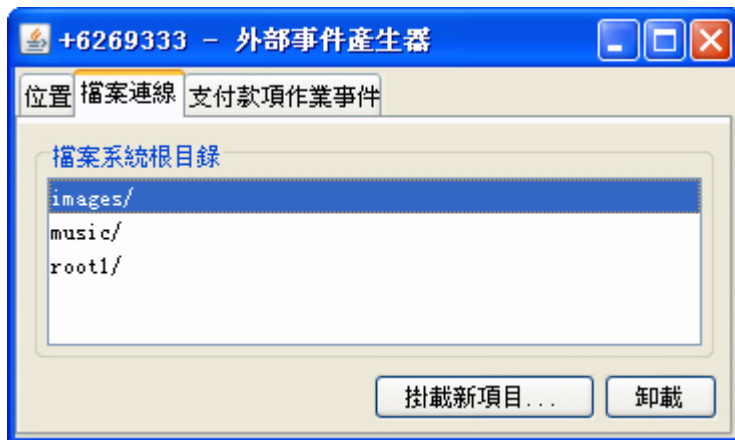
---

**備註 –** 如果同時執行同一個模擬器面板的多個實例，Sun Java™ Wireless Toolkit for CLDC 會為每個實例產生唯一的檔案路徑。例如，將第一個目錄命名為 DefaultColorPhone，將第二個目錄命名為 DefaultColorPhone1。

---

filesystem 的每個子目錄稱為一個根目錄。Sun Java™ Wireless Toolkit for CLDC 提供管理根目錄的機制。當模擬器正在執行時，從模擬器的視窗功能表中，選擇 [MIDlet] > [ 外部事件 ]。將顯示用於增加或移除根的公用程式視窗。

圖 10-1 管理檔案系統根目錄



使用 FileConnection API 可以存取已掛載的根目錄及其內容。

若要增加新的根目錄，請按一下 [ 掛載新項目 ]，並填入目錄的名稱。若要讓 FileConnection API 無法存取目錄，請在清單中選取該目錄，再按一下 [ 卸載 ]。

## 10.2 PIM API

Sun Java™ Wireless Toolkit for CLDC 模擬器將連絡人、行事曆及待辦資訊儲存成標準檔案，放在桌上型電腦的硬碟內。所有資訊會儲存於 `workdir\appdb\skin\pim` 中。此目錄由所有執行中的模擬器共用。清單分別儲存在 `contacts`、`events` 與 `todo` 目錄的子目錄中。例如，Contacts 連絡人清單就包含在以下路徑中：

---

**Windows:** `workdir\appdb\skin\pim\contacts\Contacts`

**Linux:** `workdir/appdb/skin/pim/contacts/Contacts`

---

在清單目錄中，項目儲存為 vCard (.vcs) 或 vCalendar (.vcf) 格式 (請參閱 <http://www.imc.org/pdi/>)。連絡人儲存為 vCard 格式，而行事曆與待辦事項則都儲存為 vCalendar 格式。

## 使用藍芽與 OBEX API

---

Sun Java™ Wireless Toolkit for CLDC 模擬器支援藍芽的 Java API：JSR 82。模擬器與 1.1 版規格完全相容，此規格描述如何與推送註冊表整合。JSR 82 包含兩個獨立的 API：

- 藍芽 API 提供藍芽無線網路的介面，包括裝置探索與資料交換。
- OBEX API 讓應用程式可以透過藍芽或其他通訊通道使用 Object Exchange (OBEX) 協定。

這一章描述 Sun Java™ Wireless Toolkit for CLDC 如何實作藍芽與 OBEX API。

---

### 11.1 藍芽模擬環境

Sun Java™ Wireless Toolkit for CLDC 模擬器可讓您開發及測試使用藍芽的應用程式，而無需使用真的藍芽硬體。工具組會模擬執行模擬器所需的藍芽環境。多個模擬器實例可以使用藍芽 API 互相探索，並交換資料。

如需範例，請參閱附錄 A 中的 BluetoothDemo 文件。

---

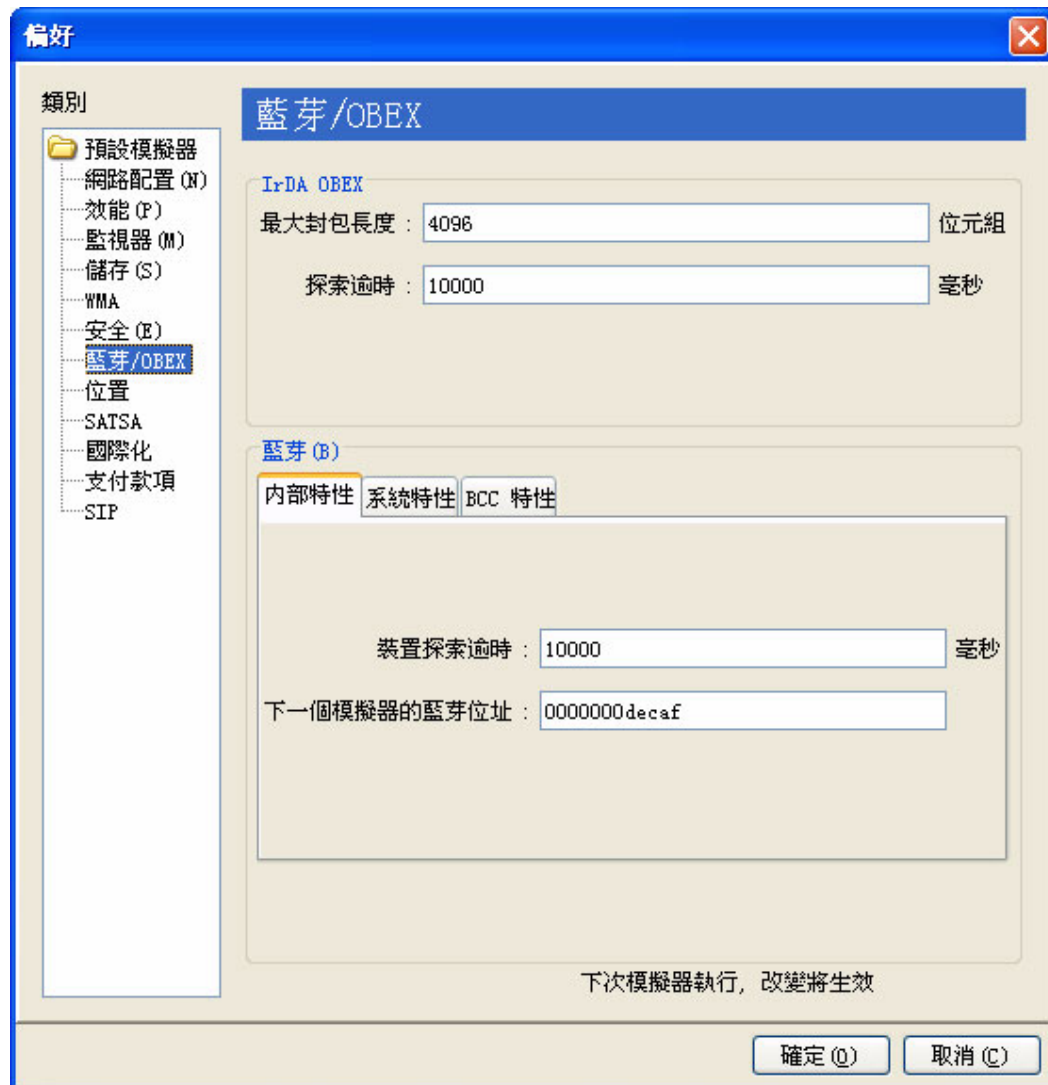
### 11.2 透過紅外線使用 OBEX

Sun Java™ Wireless Toolkit for CLDC 透過模擬的藍芽和紅外線連線來實作 OBEX 傳輸。模擬的紅外線連線遵循 IrDA 標準（請參閱 <http://www.irda.org/>）。模擬的紅外線傳輸可在多個執行中的模擬器之間進行。

## 11.3 設定 OBEX 與藍芽偏好

Sun Java™ Wireless Toolkit for CLDC 可讓您配置藍芽與 OBEX 模擬環境。選擇 [編輯] > [偏好]，並選取 [藍芽 / OBEX]，以顯示下列視窗。

圖 11-1 藍芽與 OBEX 偏好





## 11.3.1 OBEX 偏好

實際作業時，使用 IrDA 的裝置用偵聽的方式來探索其他裝置。您可以用偏好視窗 [IrDA OBEX] 區段中的 [探索逾時] 欄位，配置 Sun Java™ Wireless Toolkit for CLDC 模擬器在探索其他裝置時，等候的時間長短。請以毫秒為單位輸入值。

在 API 層級，探索逾時值會決定呼叫

`Connector.open("irdaobex://discover...")` 區段時，要等候多久之後才會傳回或丟出異常。

最大資料封包長度會影響模擬器之間，每個資料封包中傳送的資料量。資料封包值越短，產生的資料封包越多，資料封包耗用時間也越多。

## 11.3.2 藍芽內部特性

在偏好視窗的 [藍芽] 區段中，[裝置探索逾時] 是模擬器嘗試在模擬藍芽環境中尋找其他裝置時，等候的時間長短（以毫秒為單位）。

[下一個模擬器的藍芽位址] 是要指定給第一個模擬器實例的藍芽位址。模擬器的後續實例將接收自動產生的位址。

## 11.3.3 藍芽系統特性

偏好視窗的 [藍芽] 區段中的 [系統特性] 標籤包含的特性，在應用程式中使用 `javax.bluetooth.LocalDevice` 的 `getProperty()` 方法即可擷取。

有關藍芽特性的完整說明，請參閱 JSR 82 規格。

## 11.3.4 藍芽 BCC 特性

Bluetooth Control Center (BCC) 用於控制藍芽設定。有些裝置可能會提供 GUI，以自訂藍芽設定。在 Sun Java™ Wireless Toolkit for CLDC 中，使用藍芽偏好的 [BCC 特性] 標籤配置 BCC。可配置的特性如下。

表格 11-1 BCC 特性

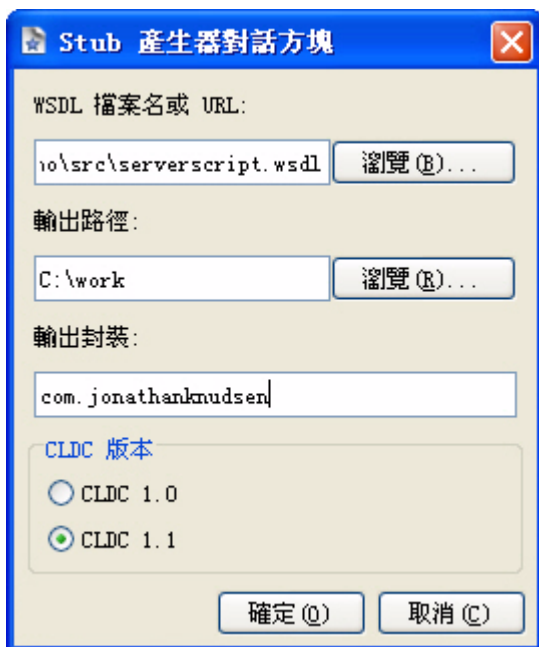
特性	描述
啟用藍芽支援	如果停用這個特性， <code>LocalDevice.getLocalDevice()</code> 會丟出 <code>BluetoothStateException</code> ，而且無法建立任何連線。其可用於測試裝置（支援 JSR 82 但是停用了藍芽功能）上應用程式的運作方式。
裝置可探索	指出其他模擬器是否能夠探索此模擬器。
易記名稱	在模擬藍芽環境中，模擬器所使用的易讀名稱。如果名稱保留空白，則模擬器不支援易記名稱功能。
加密	決定支援 (on) 還是不支援 (off) 連線加密。此外， <code>force</code> 設定表示所有連線都必須加密。如需詳細資料，請參閱有關 <code>RemoteDevice</code> 的 <code>encrypt()</code> 方法的文件。
授權	與「加密」特性類似。請參閱 <code>RemoteDevice</code> 的 <code>authorize()</code> 方法。
認證	與「加密」和「授權」類似。請參閱 <code>RemoteDevice</code> 的 <code>authenticate()</code> 方法。

## 使用 Web 服務

Sun Java™ Wireless Toolkit for CLDC 模擬器支援 J2ME Web Services Specification JSR 172。JSR 172 提供從行動應用程式存取 Web 服務的 API。同時也包含剖析 XML 文件的 API。

Sun Java™ Wireless Toolkit for CLDC 提供 Stub 產生器，能夠將存取 Web 服務的原始碼建立工作自動化。若要進入 Stub 產生器，請選擇 [檔案] > [公用程式]。按一下 [Stub 產生器]，然後按下 [啟動]。

圖 12-1 Web 服務 Stub 產生器



在 [WSDL 檔案名稱或 URL] 欄位中，輸入要存取的 Web 服務之 Webservices Description Language (WSDL) 檔案路徑。[輸出路徑] 指示儲存 Stub 檔案的位置。[輸出封裝] 指示 Stub 檔案的 Java 程式設計語言套裝軟體名稱。最後，選擇要產生 CLDC 1.0 或 CLDC 1.1 Stub。

按下 [確定] 產生 Stub 檔案。

## 使用位置 API

---

JSR 179 位置 API 可讓應用程式使用裝置的定位能力。例如，有的裝置內建全球定位系統 (GPS) 硬體。其他裝置可能能夠接收來自無線網路的定位資訊。無論基礎技術如何，位置 API 均可為定位資訊提供標準介面。

在位置 API 中，**位置提供者**可封裝定位方法並提供該裝置位置的資訊。應用程式請求提供者時，需指定所要精確度與回應時間等必要條件。如果有適當的實作，則應用程式可使用該實作取得有關裝置實體位置的資訊。

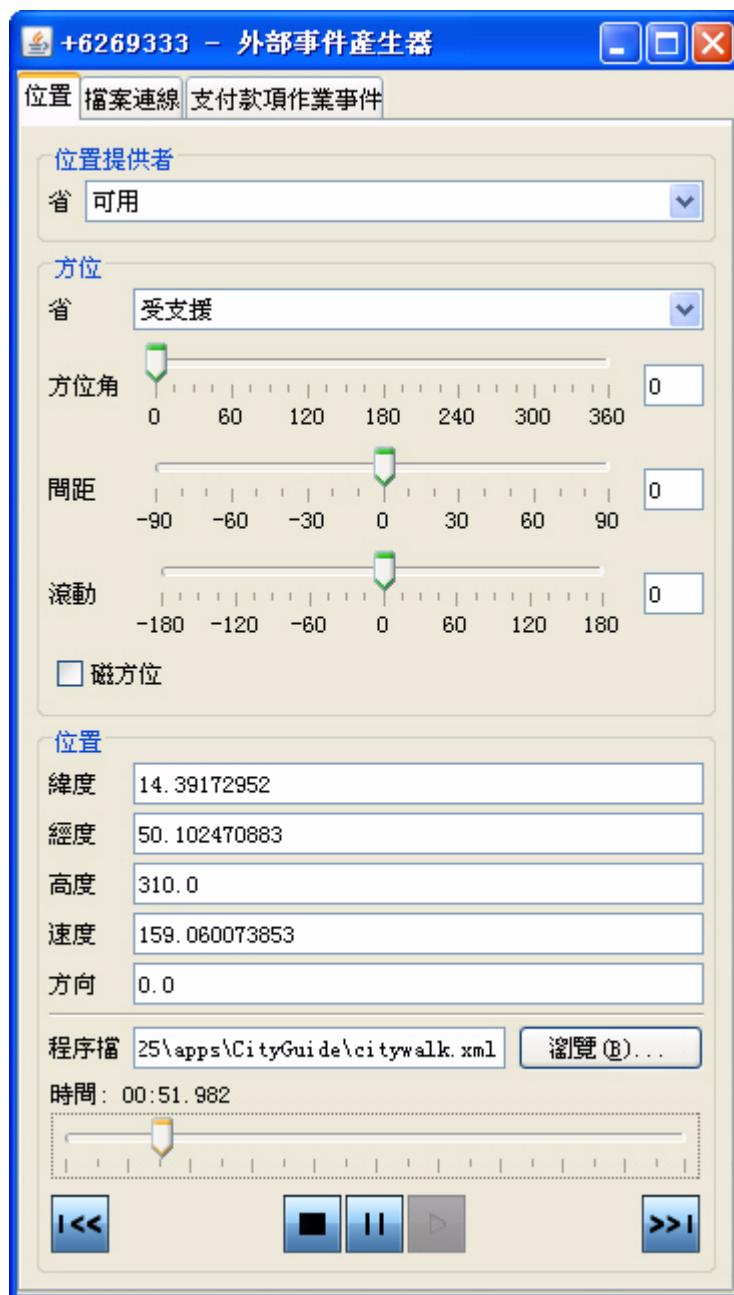
Sun Java™ Wireless Toolkit for CLDC 包含一個模擬位置提供者。您可以使用模擬器的 [外部事件] 視窗，指定模擬器所處的位置。另外，您還可以配置提供者自身的特性並管理地標資料庫。

---

### 13.1 在執行階段設定模擬器的位置

您可以在模擬器執行時指定它的模擬位置。若要執行該操作，請從模擬器視窗功能表中選擇 [MIDlet] > [外部事件]。按一下 [位置] 標籤。請參閱圖 13-1。

圖 13-1 控制模擬器中的位置



在標籤的 [ 位置 ] 區域，您可以填入緯度值、經度值、高度值、速度值以及方向值。使用位置 API 的應用程式可將這些值擷取為模擬器的位置。

如需更詳細的測試，請設定描述一段時間內運動的位置程序檔。位置程序檔是包含位置清單（稱為航點）與相關時間的 XML 檔案。Sun Java™ Wireless Toolkit for CLDC 確定模擬器的目前位置時，會在位置程序檔的點中插入位置點。例如，以下是指定了起始點 (time="0") 的簡單位置程序檔，十秒鐘後該起始點將移至一個新點：

```
<waypoints>
  <waypoint time="0"
            latitude="14" longitude="50" altitude="310" />
  <waypoint time="10000"
            latitude="14.5" longitude="50.1" altitude="215" />
</waypoints>
```

以公尺作為高度的測量單位，以毫秒作為時間的測量單位。

使用文字編輯器建立位置程序檔。按下 [ 程序檔 ] 欄位旁邊的 [ 瀏覽 ] 按鈕即可將該程序檔載入外部事件視窗。正下方即是用於播放、暫停、停止，以及移至該位置程序檔開頭與結尾的控制項。您也可以將時間滑動軸拖曳至某一特定點。

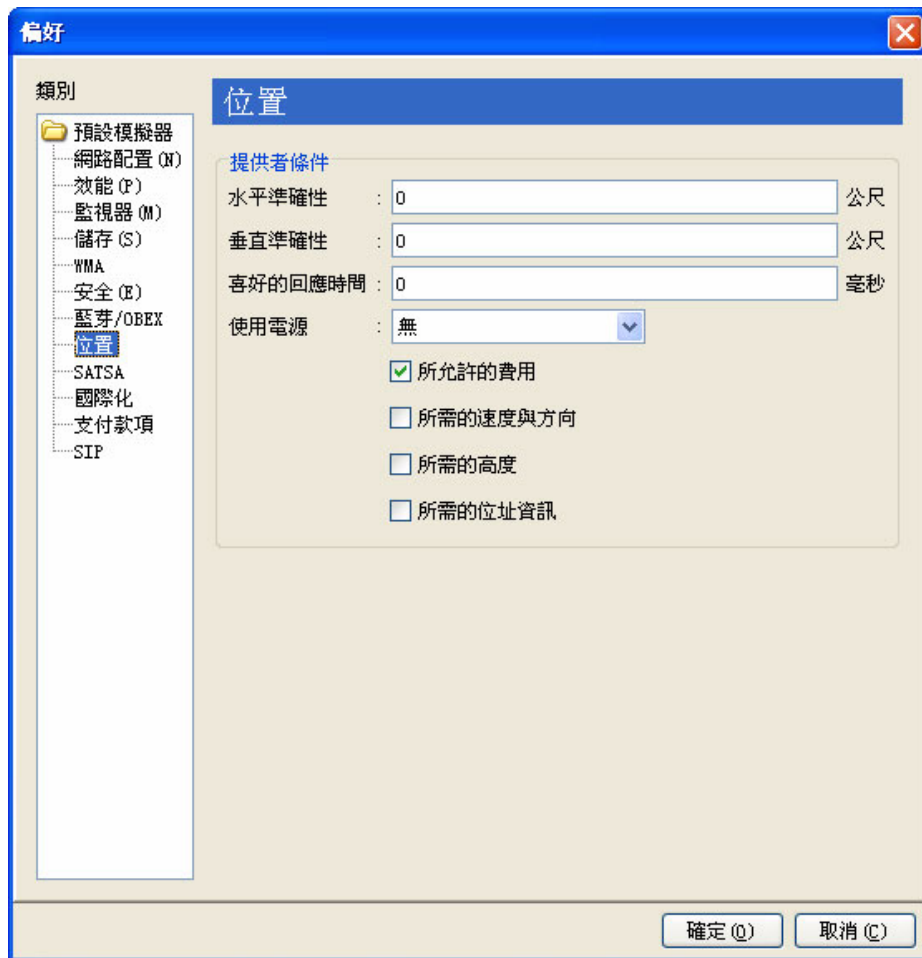
某些裝置還可以測量自己的方位。若要使您的應用程式可以使用此類資訊，請將 [ 方位 ] 方塊中的 [ 狀態 ] 欄位變更爲 [ 受支援 ] 並填入方位角、間距以及滾動值。[ 磁方位 ] 核取方塊表示方位角與間距的測量單位是與地球磁場有關還是與真北和重力有關。

若要測試您的應用程式處理異常情況的能力，請嘗試將 [ 位置提供者 ] 方塊中的 [ 狀態 ] 欄位變更爲 [ 暫時不可用 ] 或 [ 停止服務 ]。當您的應用程式嘗試擷取模擬器位置時，將丟出一個異常，您可以藉此檢查應用程式是如何回應的。

## 13.2 配置位置提供者

您可以使用偏好配置 Sun Java™ Wireless Toolkit for CLDC 的位置提供者的特性。在使用者介面中，請選擇 [ 編輯 ] > [ 偏好 ]，然後按一下 [ 位置 ]。

圖 13-2 配置位置提供者



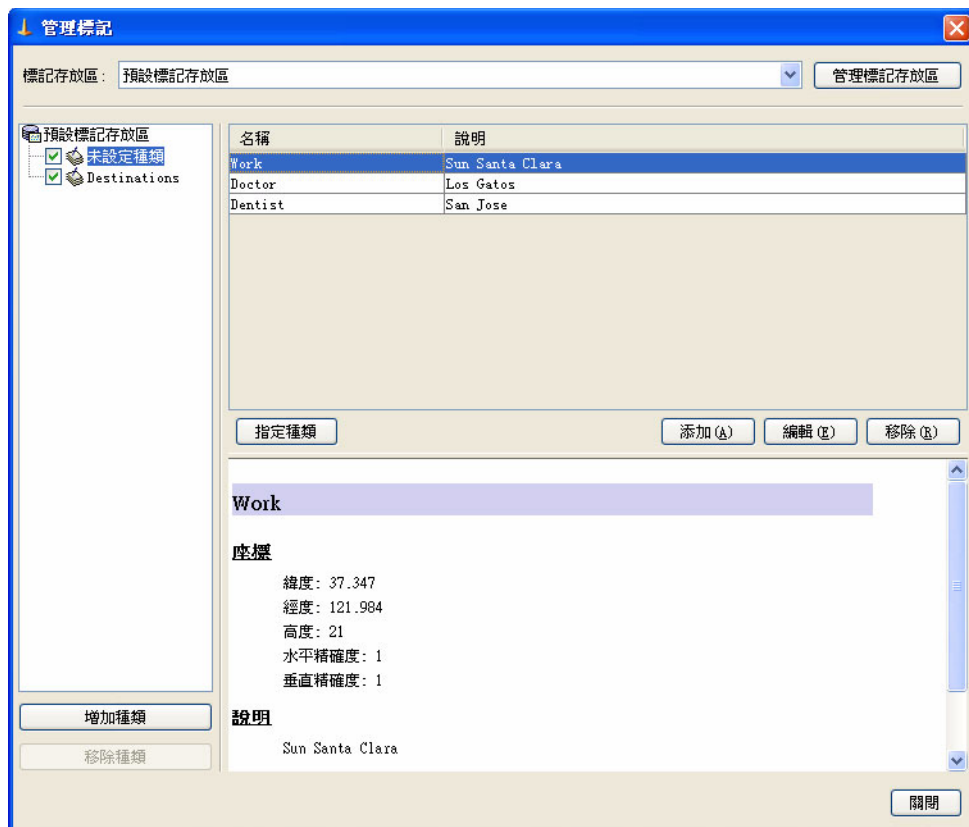
[ 位置 ] 標籤中的欄位可讓您指定工具組的內建位置提供者的特性。您在偏好中指出的特性與用來請求位置提供者的 Criteria 類別應用程式相對應。



## 13.3 設定地標

Sun Java™ Wireless Toolkit for CLDC 模擬器和許多實際裝置一樣，包含一個地標存放區系統。地標存放區是地名及其他資訊的集合。若要管理地標存放區，請從功能表中選擇 [ 檔案 ] > [ 公用程式 ]，然後選取 [ 管理地標 ] 並按一下 [ 啟動 ]。

圖 13-3 地標管理員



地標管理員可顯示單一地標存放區的內容。JSR 179 最少需要一個地標存放區，亦稱為預設存放區。若要選取其他地標存放區或建立新的地標存放區，請從視窗頂部的 [ 地標存放區 ] 組合方塊中選取某個選項。

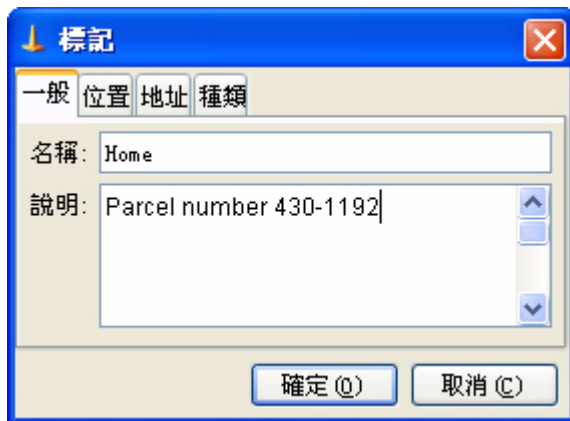
您可以透過按一下 [ 管理地標存放區 ] 來增加或移除地標存放區。無法重新命名地標存放區。

地標可以與特定於某地標存放區的**種類**建立關聯。目前地標存放區的種類顯示在視窗的左窗格中。您可以使用清單底部的按鈕增加或移除種類。如果您只希望看到相符的地標，請取消核取一個或多個種類。您還可以核取**未設定種類**以顯示不具有相關聯之種類的地標。

地標管理員的右側窗格中列出了目前地標存放區中的地標。按一下某個地標以在右側窗格底部查看其詳細資訊。

若要增加新地標，請按一下**增加**，然後視需要填寫欄位。按一下**編輯**以變更目前所選的地標。最後按下**移除**以移除目前所選的地標。

■ 13-4 增加或編輯地標



您也可以使用主視窗中的**指定種類**來指定地標的種類。

## 使用 SATSA

Security and Trust Services API (SATSA) 讓小型裝置上執行的應用程式，擁有智慧卡存取和加密功能。JSR 177 (SATSA 規格) 定義了四個不同的 API 做為選擇性套裝軟體：

- **SATSA-APDU** - 可讓應用程式使用低階協定與智慧卡應用程式通訊。
- **SATSA-JCRMI** - 使用遠端物件協定，提供另一種與智慧卡應用程式通訊的方式。
- **SATSA-PKI** - 可讓應用程式使用智慧卡為資料加上數位簽名並管理使用者憑證。
- **SATSA-CRYPTO** - 一般用途的加密 API，支援訊息摘要、數位簽名和密碼。

Sun Java™ Wireless Toolkit for CLDC 模擬器完全支援 SATSA。這一章描述如何使用 Sun Java™ Wireless Toolkit for CLDC，以在您自己的應用程式中使用 SATSA。

如需 SATSA 和在小型裝置上使用智慧卡的一般簡介，請參閱「SATSA 開發人員指南」，該指南可在

<http://java.sun.com/j2me/docs/satsa-dg/> 上取得。

Sun Java™ Wireless Toolkit for CLDC 包含 Java Card Platform Simulator，您可將其用於模擬 Sun Java™ Wireless Toolkit for CLDC 模擬器槽中的智慧卡。Java Card 平台模擬器位於：

---

**Windows:** `toolkit\bin\cref.exe`

**Linux:** `toolkit/bin/cref`

---

下文中，我們將其簡稱為 cref。

如果需要開發自己的 Java Card 應用程式，請下載 Java Card Development Kit，其可以在 <http://java.sun.com/products/javacard/> 上取得。

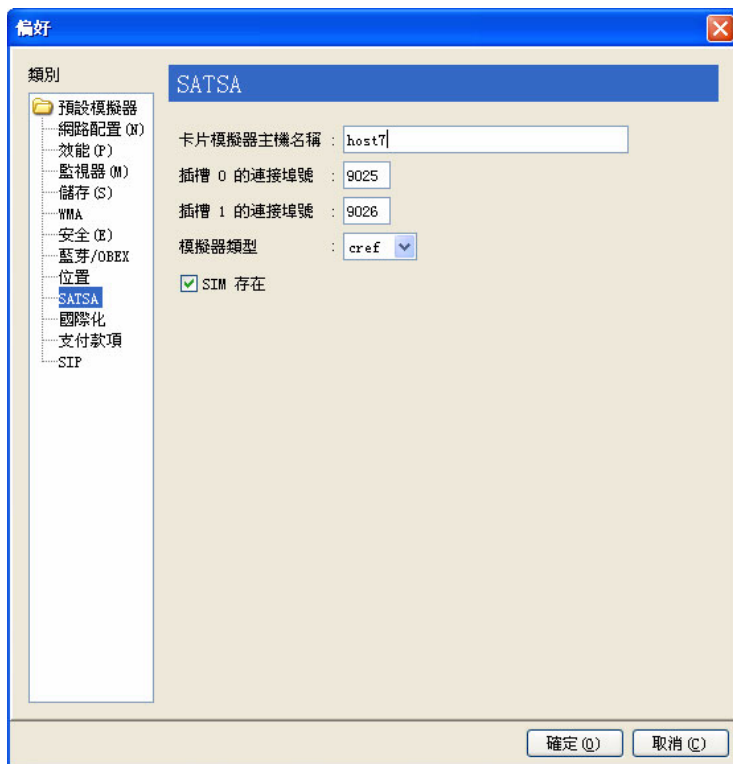
## 14.1 模擬器中的卡片槽

實際 SATSA 裝置可能具有用於插接智慧卡的一個或多個插槽。使用 SATSA 與智慧卡通訊的應用程式需要指定卡片槽和智慧卡應用程式。

Sun Java™ Wireless Toolkit for CLDC 模擬器不是實際裝置，因此沒有智慧卡的實體插槽，而是使用通訊端協定與智慧卡應用程式通訊。通訊端的另一端可能是智慧卡模擬器，或者可能是與實際智慧卡硬體通訊的代理協定。

Sun Java™ Wireless Toolkit for CLDC 模擬器包含兩個模擬的智慧卡槽。每個槽都有一個相關聯的通訊端，代表和智慧卡應用程式通訊的協定之一端。您可以為每個槽設定通訊端連接埠號。請選擇 [編輯] > [偏好]，然後按一下 [SATSA] 標籤。預設連接埠為 9025 用於槽 0 以及 9026 用於槽 1。

圖 14-1 設定智慧卡槽的連接埠號



---

## 14.2 使用 Java Card 平台模擬器

使用 Sun Java™ Wireless Toolkit for CLDC 測試 SATSA 應用程式的基本步驟如下：

1. 使用 **Java Card** 平台應用程式啟動 `cref`。
2. 啟動 **Sun Java™ Wireless Toolkit for CLDC** 模擬器。

當 SATSA 應用程式嘗試與智慧卡通訊時，會使用通訊端連線與 `cref` 通訊。

因此啟動 `cref` 時所用的連接埠號，一定要與您在 Sun Java™ Wireless Toolkit for CLDC 偏好中指定的其中一個槽連接埠號相同。

例如，您可以使用以下的指令行，在具有預先建立記憶體影像的連接埠 9025 上執行 `cref`：

```
cref -p 9025 -i memory_image.eeprom
```

Sun Java™ Wireless Toolkit for CLDC 包含一個示範應用程式 Mohair，可說明如何使用 SATSA。如需有關執行 Mohair 的詳細簡介，請參閱[附錄 A](#)。

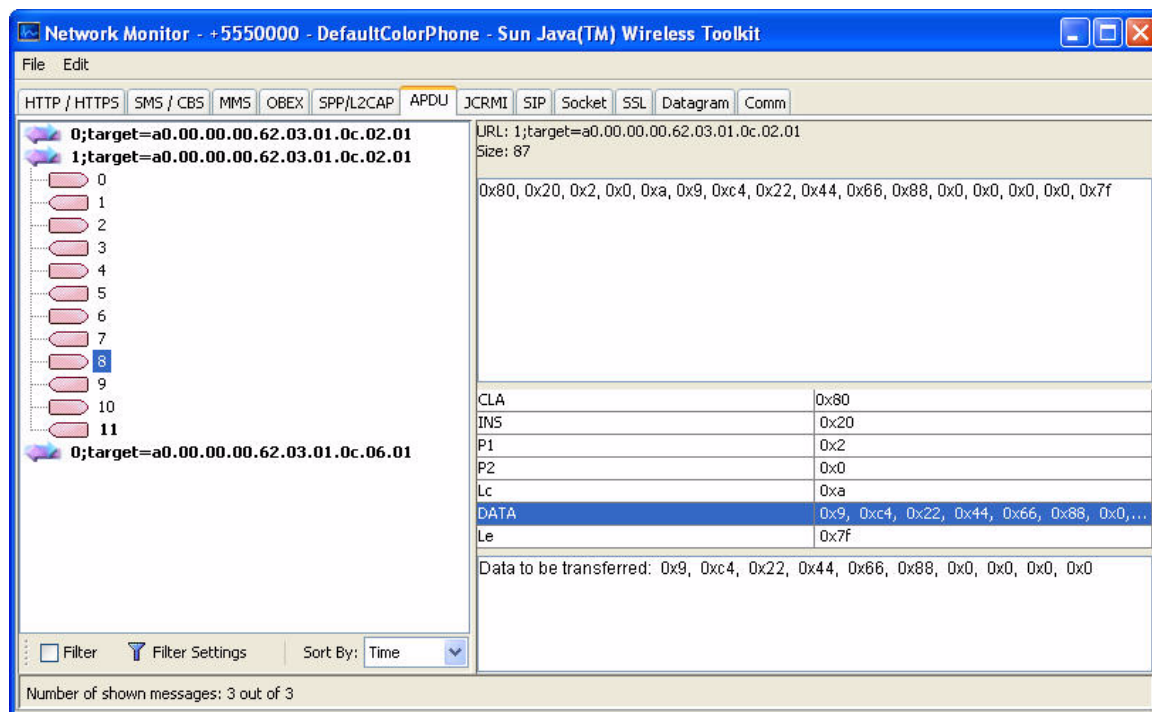
---

## 14.3 使用具有 SATSA 的網路監視器

Sun Java™ Wireless Toolkit for CLDC 可以在網路監視器中顯示與模擬智慧卡應用程式進行交換的資料。網路監視器會顯示模擬器和智慧卡模擬器交換的 Application Protocol Data Unit (APDU)，也可以顯示使用 Java Card Remote Method Invocation (Java Card RMI) 協定進行交換的資料。網路監視器中的 APDU 和 JCRMI 標籤可以顯示與智慧卡進行交換的資料。

網路監視器可以剖析每個 APDU，並視需要顯示請求和回應的欄位。

圖 14-2 在網路監視器中檢視 APDU



## 14.4 調整存取控制

存取控制權限和 PIN 特性可以在文字檔中指定。當第一個 APDU 或 Java Card RMI 連線建立後，實作會讀取 *workdir\appdb* 目錄中 *acl\_slot-number* 的 ACL 和 PIN 資料。例如，槽 0 的存取控制檔為 *workdir\appdb\acl\_0*。如果缺少此檔案或檔案包含錯誤，則會停用此槽的存取控制驗證。

此檔案可以包含有關 PIN 特性和應用程式權限的資訊。

## 14.4.1 指定 PIN 特性

PIN 特性由存取控制檔中的 `pin_data` 記錄表示。

```
pin_data {
    label string
    id number
    type      bcd | ascii | utf | half-nibble | iso
    min       minLength -
    stored    storedLength
    max       maxLength
    reference byte
    pad       byte - optional
    flag      case-sensitive | change-disabled |
              unblock-disabled | needs-padding |
              disable-allowed | unblockingPIN
}
```

## 14.4.2 指定應用程式權限

應用程式權限要在存取控制檔 (acf) 記錄中定義。

```
acf AID f 由空格分隔的數字 {
    ace {
        root CA name
        ...
        apdu {
            八個由空格分隔的數字
            ...
        }
        ...
        jcrmi {
            classes {
                classname
                ...
            }
            hashModifier string
            methods {
                方法名稱和簽名
                ...
            }
        }
        ...
        pin_apdu {
            id number
        }
    }
}
```

```

        verify | change | disable | enable | unblock
        四個十六進制的數字
        ...
    }
    ...
    pin_jcrmi {
        id number
        verify | change | disable | enable | unblock
        方法名稱和簽名
        ...
    }
    ...
}

```

acf 記錄是一個存取控制檔。acf 之後的 AID 會識別應用程式。缺少 AID 表示此項目將套用至所有應用程式。acf 記錄可以包含 ace 記錄。如果沒有 ace 記錄，則對應用程式的存取受到此 acf 的限制。

ace 記錄是一個存取控制項目。它可以包含 root、apdu、jcrmi、pin\_apdu 以及 pin\_jcrmi 記錄。

root 記錄包含一個 CA 名稱。若使用此 CA 發出的憑證來授權 MIDlet 套件，則此 ace 會授予對此 MIDlet 的存取。缺少 root 欄位代表 ace 將套用至所有已識別的對象。一行描述一個主體。該行必須僅包含文字 root 和主體名稱，例如：

```
root CN=thelhost;OU=JCT;O=dummy CA;L=Santa Clara;ST=CA;C=US
```

apdu 或 jcrmi 記錄描述了 APDU 或 Java Card RMI 權限。缺少權限記錄代表允許所有作業。

APDU 權限包含一組或多組八個十六進制值序列，以空格分隔。前四個位元組描述了 APDU 指令，其餘四個位元組為遮罩，例如：

```
apdu {
    0 20 0 82 0 20 0 82
    80 20 0 0 ff ff 0 0
}

```



Java Card RMI 權限包含有關雜湊修飾符（選擇性）、類別清單和方法清單（選擇性）的資訊。如果方法清單空白，則允許應用程式呼叫類別清單中介面的所有遠端方法，例如：

```
jcrmi {
    classes {
        com.sun.javacard.samples.RMIDemo.Purse
    }
    hashModifier zzz
    methods {
        debit(S)V
        setAccountNumber([B)V
        getAccountNumber()[B
    }
}
```

所有數字均為十六進制。表格、空格、CR 和 LF 符號用作分隔符號。在符號 { 和 } 前後可以沒有分隔符號。

pin\_apdu 和 pin\_jcrmi 記錄包含 PIN 輸入方法，也就是 PIN 識別碼和 APDU 指令標頭或遠端方法名稱。

### 14.4.3 存取控制檔範例

```
pin_data {
    label    Unblock pin
    id       44
    type     utf
    min      4
    stored   8
    max      8
    reference 33
    pad      ff
    flag     needs-padding
    yflag    unblockingPIN
}
pin_data {
    label    Main pin
    id       55
    type     half-nibble
    min      4
    stored   8
    max      8
    reference 12
    pad      ff
}
```

```

flag      disable-allowed
flag      needs-padding
}

acf a0 0 0 0 62 ff 1 {
  ace {
    root CN=thehost;OU=JCT;O=dummy CA;L=Santa Clara;ST=CA;C=US

    pin_jcrmi {
      id 55
      verify enterPIN([B)S
      change changePIN([B[B)S
      disable disablePIN([B)S
      enable enablePIN([B)S
      unblock unblockPIN([B[B)S
    }
  }
}

acf a0 0 0 0 62 ee 1 {
  ace {
    root CN=thehost;OU=JCT;O=dummy CA;L=Santa Clara;ST=CA;C=US

    pin_apdu {
      id 55
      verify 1 2 3 1
      change 4 3 2 2
      disable 1 1 1 3
      enable 5 5 5 4
      unblock 7 7 7 5
    }
  }
}

acf a0 0 0 0 62 3 1 c 8 1 {
  ace {
    root CN=thehost;OU=JCT;O=dummy CA;L=Santa Clara;ST=CA;C=US

    jcrmi {
      classes {
        com.sun.javacard.samples.RMIDemo.Purse
      }
      hashModifier xxx
      methods {
        setAccountNumber([B)V
        getBalance()S
        credit(S)V
      }
    }
  }
}

```

```

}
ace {
    jcrmi {
        classes {
            com.sun.javacard.samples.RMIDemo.Purse
        }

        debit(S)V
        getAccountNumber()[B
    }
}

}

acf a0 00 00 00 62 03 01 0c 02 01 {
    ace {
        root CN=thehost;OU=JCT;O=dummy CA;L=Santa Clara;ST=CA;C=US
        apdu {
            0 20 0 82 0 20 0 82
            80 20 0 0 ff ff 0 0
        }
        apdu {
            80 22 0 0 ff ff 0 0
        }
    }
}

acf a0 00 00 00 62 03 01 0c 02 01 {
    ace {
        apdu {
            0 20 0 82 ff ff ff ff
        }
    }
}

acf a0 00 00 00 62 03 01 0c 06 01 {
    ace {
        apdu {
            0 20 0 82 ff ff ff ff
        }
    }
}

```



## 使用 SIP

---

Sun Java™ Wireless Toolkit for CLDC 支援具有代理伺服器、代理登錄以及網路監視器支援的 SIP API for J2ME (JSR 180)。

Session Initiation Protocol (SIP) 由 RFC 3261 定義，可在 <http://www.ietf.org/rfc/rfc3261.txt> 上取得。

SIP 為應用程式提供設定通訊的標準方式。應用程式決定實際發生的通訊。SIP 可用於設定即時訊息傳送、文字聊天、語音聊天、視訊會議或其他類型的階段作業。

---

### 15.1 瞭解代理登錄和代理伺服器

SIP 代理登錄可讓用戶端應用程式將使用者名稱與特定網路位址建立關聯。實際上，註冊為使用者提供了一個表明「我在這裡！」的方式。

SIP 代理伺服器實際上只是較大代理伺服器網路的進入點。到達一個代理伺服器的 SIP 訊息會路由至適當目標，該目標通常為另一個代理伺服器或端點，例如桌上型電腦或行動裝置。雖然 SIP 訊息可以直接在裝置間傳送，但是通常要透過代理伺服器來路由。

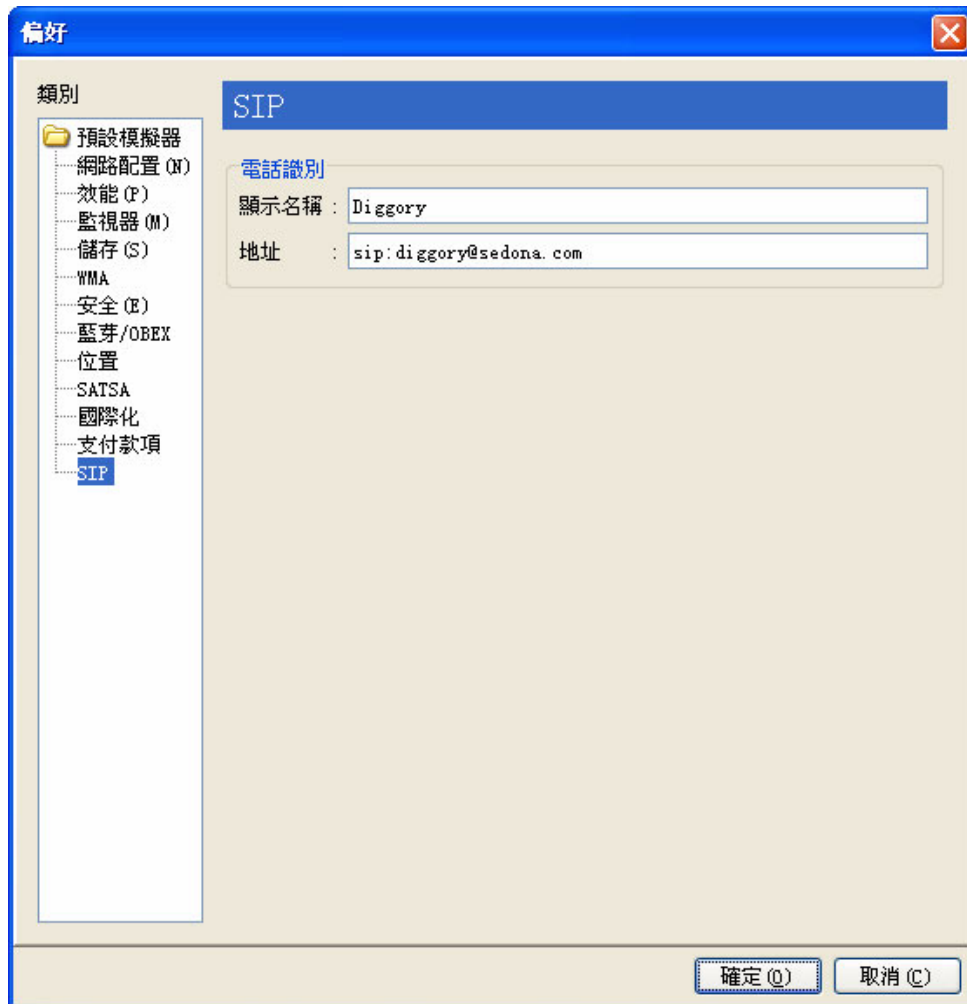
例如，假設 Diggory 想要與 Polly 進行視訊會議。Polly 正在路上，她的行動電話會傳送一則訊息給代理登錄；該代理登錄將她的名字和行動電話網路位址建立關聯。當 Diggory 嘗試建立與 Polly 的視訊會議時，他的應用程式會使用 SIP 要求代理登錄提供 Polly 目前的網路位置。

Sun Java™ Wireless Toolkit for CLDC 包含一個非常簡單的 SIP 代理伺服器和代理登錄伺服器，讓您用來測試使用 SIP API 的應用程式。您也可以將工具組配置為使用外部代理伺服器和代理登錄伺服器。

## 15.2 SIP 設定

若要調整 Sun Java™ Wireless Toolkit for CLDC 的 SIP 環境設定，請選擇 [ 編輯 ] > [ 偏好 ] 並按一下 [SIP]。

圖 15-1 SIP 設定



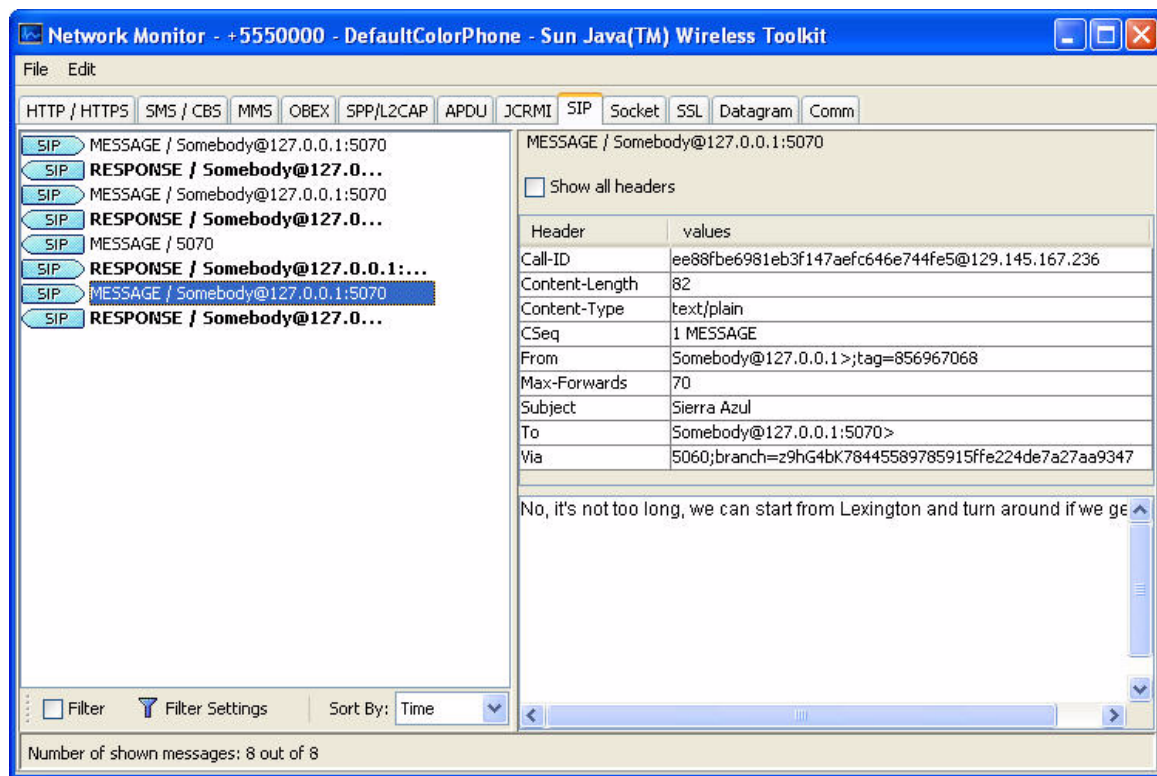
[ 顯示名稱 ] 和 [ 地址 ] 欄位可以分別設定系統特性 `sip.display.name` 和 `sip.address`。應用程式可以把這些系統特性，當成擷取與裝置相關聯之身份的標準方式。

## 15.3 網路監視器中的 SIP 流量

網路監視器可以記錄使用 SIP API 傳送和接收的網路資料。網路模擬器在[第 5 章](#)中有詳盡描述。您可以使用網路監視器追蹤傳入 / 傳出模擬器的 SIP 訊息。

按一下 [SIP] 標籤即可看到 SIP 訊息。SIP 訊息顯示於網路監視器左側窗格中。按一下訊息即可在右側窗格中看到詳細資訊。

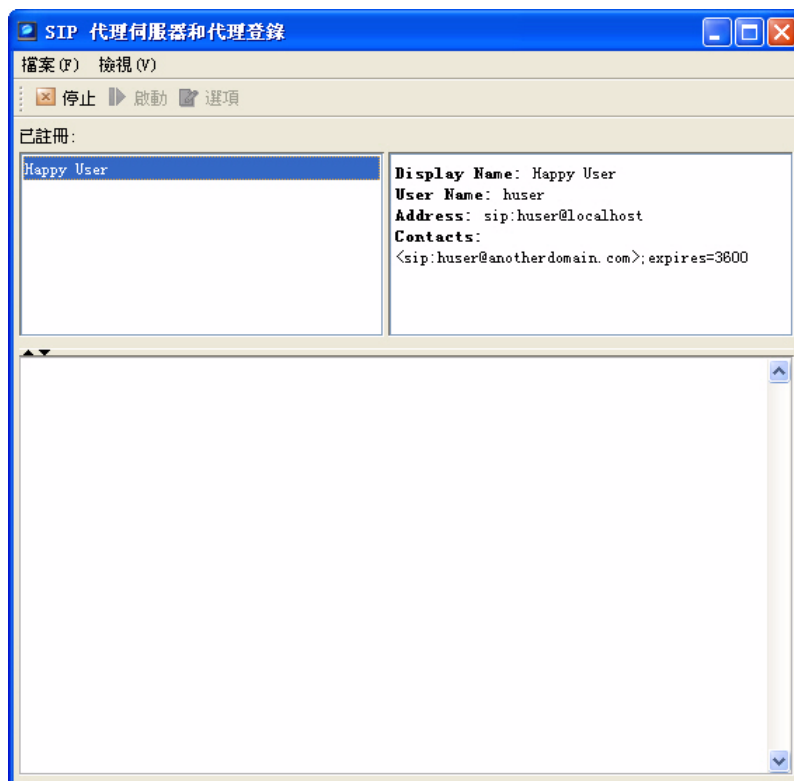
圖 15-2 網路監視器中的 SIP 訊息



## 15.4 SIP 代理伺服器和代理登錄

Sun Java™ Wireless Toolkit for CLDC 提供了一個簡單的 SIP 代理伺服器和代理登錄，讓您建立 SIP 應用程式時更輕鬆。若要啟動伺服器，請選擇 [ 檔案 ] > [ 公用程式 ]。從清單中選取 [ 啟動 SIP 伺服器 ]，並按下 [ 啟動 ]。就會顯示 SIP 伺服器主控台視窗。

圖 15-3 SIP 伺服器主控台



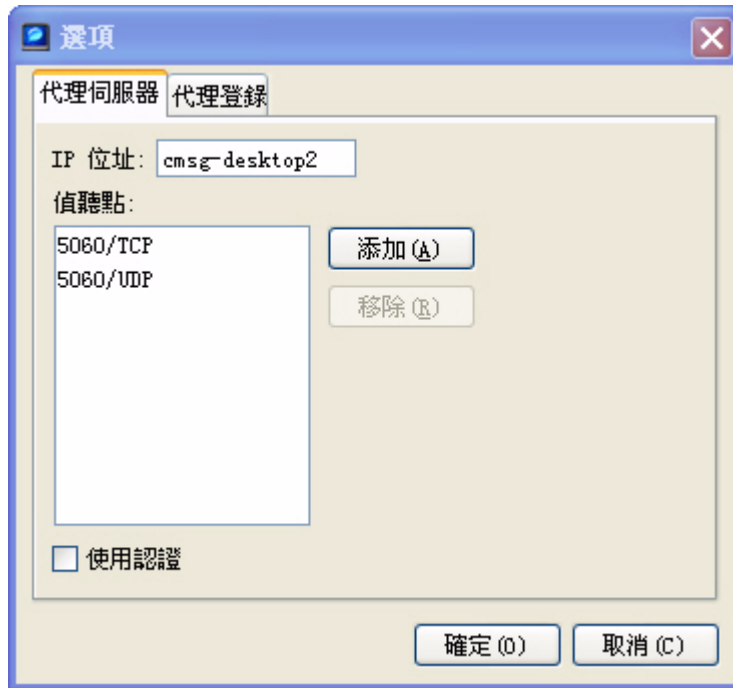
若要停止伺服器，請按一下 [ 停止 ]。若要啟動伺服器，請按一下 [ 啟動 ]。

伺服器在執行時，左上窗格會顯示代理登錄已知的所有使用者。按一下使用者名稱即可在右上窗格中看到使用者的詳細資訊。視窗的底部窗格是主控台，可顯示代理伺服器傳送和接收的 SIP 訊息。

您可以在沒有執行伺服器的時候調整伺服器選項。停止伺服器並按一下 [ 選項 ] 即可看到 [ 選項 ] 視窗 (請參閱圖 15-4)。



圖 15-4 設定代理伺服器選項



在【代理伺服器】標籤上，您可以設定伺服器偵聽內送訊息所使用的 IP 位址和連接埠。請注意 5060（預設偵聽點）是 SIP 代理伺服器的常用連接埠。如果您在多重使用者環境中工作，則其他使用者可能也在使用該連接埠，而您可能無意中連接至他人的 SIP 伺服器實例（SIP 伺服器不具有任何認證機制，TCP/IP 連接埠可以自由存取）。如果發生此種情況，您必須指定其他連接埠。

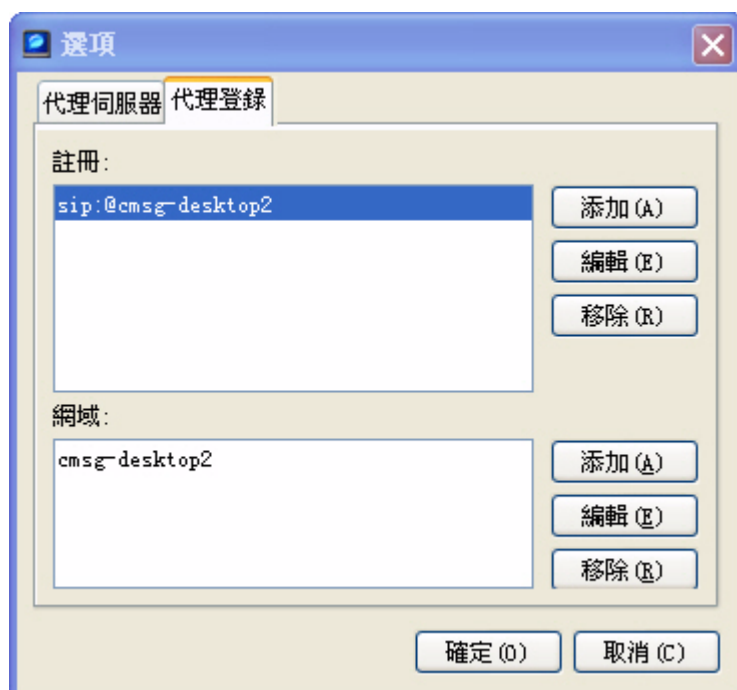
按一下【增加】指定多個連接埠及其類型。選取一個連接埠並按一下【移除】即可移除偵聽連接埠。

核取【使用認證】可強制要連線的用戶端通過伺服器認證。使用的機制為摘要式認證，RFC 3261 的第 22.4 節已描述此認證。SIP 摘要式認證與 HTTP 摘要式認證幾乎完全相同。

在【代理登錄】標籤上，您可以設定代理登錄已知的使用者和網域。頂部清單包含當 SIP 伺服器啟動時自動註冊的 SIP 使用者。您可以增加新的使用者、編輯現有的使用者或移除使用者。

另外，您可以調整由此代理登錄管理的網域的清單。按下【增加】可增加網域、按下【編輯】可編輯現有的網域名稱或按下【移除】移除網域。

圖 15-5 設定代理登錄選項



## 使用 Payment API

---

JSR 229 (Payment API) 可讓應用程式代表其使用者執行支付款項。Payment API 透過支付款項配接卡可支援不同的支付款項機制。實作 Payment API 的裝置具有一個或多個配接卡。MIDlet 套裝使用描述元屬性指定可使用的支付款項介面類型。

Sun Java™ Wireless Toolkit for CLDC 的模擬器採用之支付款項介面範例，可模擬 Premium Priced SMS (PPSMS) 與信用卡支付款項，以實作 Payment API。另外，工具組可讓您輕鬆設定 MIDlet 描述元與 JAR 檔案清單的必要屬性。最後，支付款項主控台可讓您輕鬆追蹤應用程式支付的款項，或支付款項作業失敗事件。

由於 Payment API 與佈建及外部裝置支付款項機制密切相關，而且只有在可信任的保護網域中才能順利支付款項，因此請務必使用工具組的 [經由 OTA 執行] 機制對 Payment API 應用程式進行測試與除錯。如需詳細資訊，請參閱第 2 章。

---

### 16.1 支付款項的專案設定

若要調整專案的支付款項屬性，請按一下 [設定]，然後選取 [支付款項] 圖示。

圖 16-1 支付款項設定

專案 "JBricks" 的設置

## 支付款項

API 選擇

必需的

選擇性的

用戶定義的

MIDlets

推送註冊表

外部 API

權限

內容處理器

支付款項

### 一般

版本: 1.0

更新戳記: 2004-08-12T13:30:00Z

更新 URL: http://localhost/jbricks/bin/jbricks.jsp

快取記憶體:

### 除錯

☐ 示範模式 ☐ 無法初始化 ☐ 無法 IO ☐ 隨機測試 ☐ 無配接卡

作業事件已遺失: 0

自動請求模式:

### 功能

功能	數值
Pay-Feature-0	0
Pay-Feature-1	1
Pay-Feature-2	2
Pay-Feature-3	3

增加 移除

### 提供者

提供者

SONERA

VISA

RADIOG

DNSDNA

MASTERCARD

增加 編輯 移除

確定 (O) 取消 (C)

JSR 229 (Payment API 規格) 詳細說明了欄位與值。

[ 一般 ] 方塊包含有關正在使用的 Payment API 版本以及何處可找到支付款項更新的資訊。若要測試，您可以指定 localhost URL ( 如螢幕擷取畫面所示 )，以直接從專案目錄中擷取更新檔案。

[ 除錯 ] 方塊包含可協助測試應用程式的選項。Payment API 規格中對每個選項都作了解釋。

[ 功能 ] 方塊列出應用程式可以具有的功能。這些功能對應到每個提供者項下所列的價格資訊。您可以使用 [ 增加 ] 與 [ 移除 ] 按鈕修改功能清單。

[ 提供者 ] 方塊列示了可用於此應用程式的特定支付款項提供者。要支付款項時，模擬器 ( 或裝置 ) 會將其中一個可用的支付款項介面，與該應用程式所列出的其中一個提供者配對。您可以使用 [ 增加 ]、[ 編輯 ] 與 [ 移除 ] 按鈕修改提供者清單。如果您增加或編輯某提供者，將顯示如下視窗。

圖 16-2 編輯支付款項提供者



**支付款項提供者設定**

**提供者資訊**

提供者名稱: RADIOG

支援的配接卡: PPSMS

貨幣: EUR

支付款項特定資訊: 747,88

**價格資訊**

標記	數值
Pay-RADIOG-Tag-0	1.35,5550000, 1_LIFE
Pay-RADIOG-Tag-1	2.75,5550000, 3_LIVES, 2
Pay-RADIOG-Tag-2	2.05,5550000, 1_LEVEL
Pay-RADIOG-Tag-3	4.05,5550000, 3_LEVELS, 2

確定 取消

Payment API 規格中亦詳細說明了這些欄位。

在 [ 價格資訊 ] 方塊中，每一行都是一個已定義的支付款項功能。若要編輯價格標籤的值，請在 [ 值 ] 欄連按兩下相應的儲存格。

---

## 16.2 直接編輯支付款項屬性

支付款項屬性儲存在副檔名為 .jpp 的支付款項更新檔案中。請閱讀該規格以瞭解全部詳細資訊。Sun Java™ Wireless Toolkit for CLDC 提供了一個公用程式，讓您無論專案設定為何，均可輕鬆編輯支付款項更新檔案。

若要執行公用程式，請選擇 [ 檔案 ] > [ 公用程式 ]，然後選取 [ 支付款項編輯 ] 對話方塊並按下 [ 啟動 ]。系統將提示您選取想要編輯的支付款項更新檔案。選擇檔案後，將顯示一個與專案設定的 [ 支付款項 ] 區段看起來幾乎一樣的視窗。支付款項編輯公用程式不包含除錯設定。

編輯支付款項更新設定，並在完成後按下 [ 確定 ]。由於支付款項更新檔案是使用加密簽署的，因此工具組將顯示金鑰清單，以供您簽署該檔案。選取您要使用的金鑰，然後按下 [ 簽署支付款項更新檔案 ]。

---

## 16.3 支付款項偏好

若要調整工具組的 Payment API 設定，請選擇 [ 編輯 ] > [ 偏好 ]，然後按一下 [ 支付款項 ]。

圖 16-3 設定支付款項偏好



行動電話國家 / 地區代碼 (簡稱 MCC) 與行動網路代碼 (簡稱 MNC) 將與 PPSMS 支付款項提供者配合使用。MCC 與 MNC 兩者搭配即可識別裝置所用的無線通信業者。進行支付款項時，MCC 與 MNC 用於從專案設定中的提供者清單中尋找相符提供者。由於模擬器並非實際裝置，因此您可填入適當的 MCC 與 MNC 值，以模擬通信業者。請參閱 Payment API 規格以瞭解更多詳細資訊。

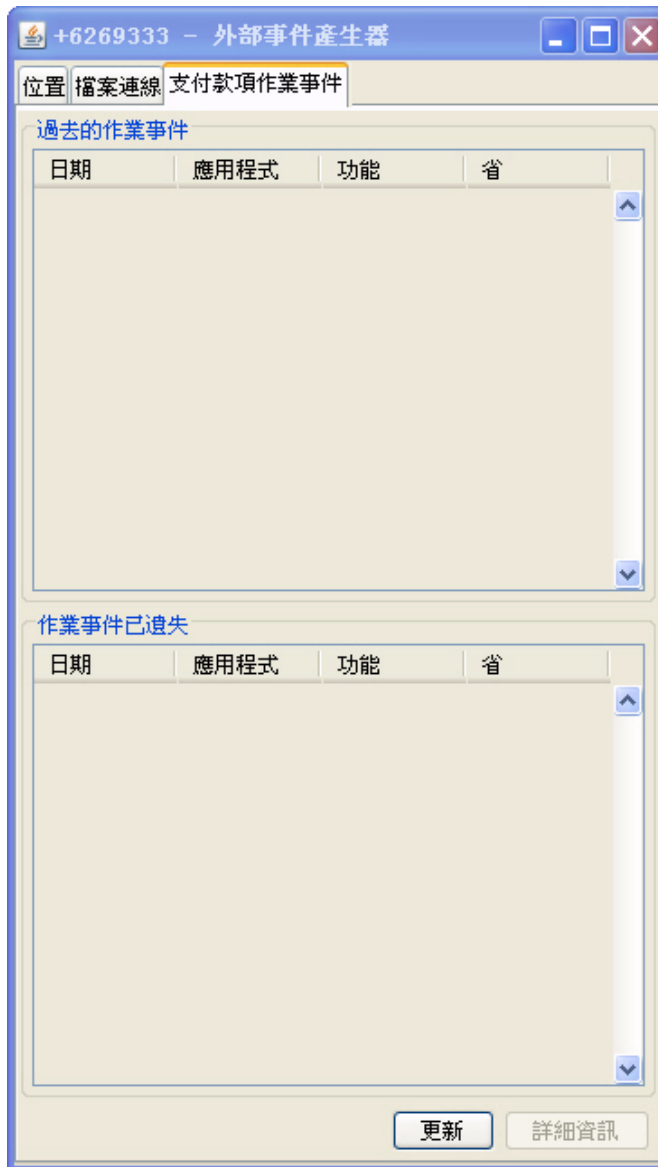
【過去作業事件數量限制】決定模擬器中所記錄的過去作業事件筆數。這將影響外部事件視窗顯示的清單長度 (如下所述)，以及應用程式自身可擷取的過去作業事件數量。

最後，【主控台電話號碼】決定支付款項主控台的模擬電話號碼，將在下文中介紹。

## 16.4 檢視作業事件歷程

模擬器將如同實際裝置一樣追蹤支付款項作業事件。若要查看作業事件歷程，請從模擬器視窗功能表中選擇 [MIDlet] > [外部事件]。按一下 [支付款項作業事件] 標籤。

圖 16-4 檢視支付款項作業事件



支付多次款項後，請按一下【更新】以更新清單。選取某作業事件，然後按一下【詳細資訊】以在獨立視窗中查看全部詳細資訊。

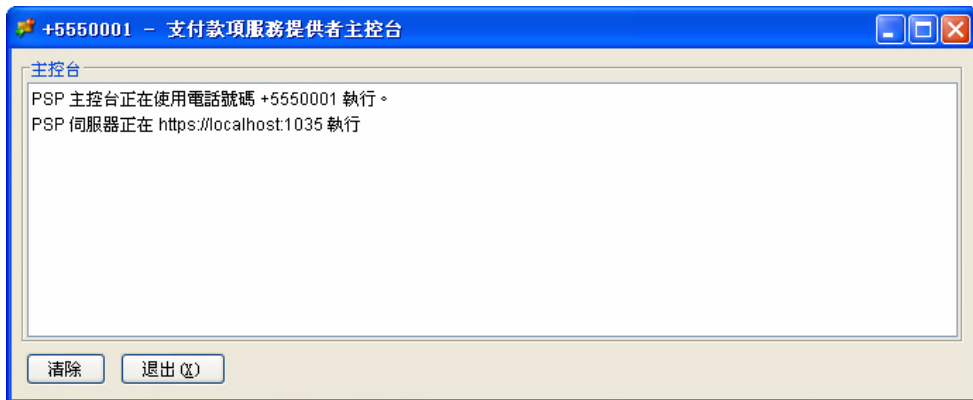
外部事件視窗僅顯示經由 OTA 執行模擬器時進行的作業事件。儘管無需使用 OTA 便可完成作業事件，但這些作業事件卻不會顯示出來。若要以最逼真的方式模擬支付款項，請務必使用【經由 OTA 執行】測試應用程式。



## 16.5 監視支付款項

Sun Java™ Wireless Toolkit for CLDC 提供支付款項主控台，可讓您輕鬆查看通過範例支付款項介面的支付款項作業。若要啟動支付款項服務提供者主控台，請選取 [ 檔案 ] > [ 公用程式 ] > [ 支付款項主控台 ]。

圖 16-5 支付款項主控台



另外，您可以使用網路監視器檢視作業事件。信用卡作業事件使用 HTTPS 完成，而 PPSMS 作業事件使用 SMS。如需網路監視器的完整描述，請參閱第 5 章。



## 使用 Mobile Internationalization API

---

JSR 238 (Mobile Internationalization API) 設計用於以多種語言顯示並在多個國家或地區使用的應用程式。國家或地區及語言的組合稱為**語言環境**。

JSR 238 的核心概念是**資源**，它可以是適於某特定語言環境的字串、影像或其他物件。例如，將在歐洲發行的應用程式所包含的資源，可能適合住義大利且說義大利語的使用者、住瑞士但說義大利語的使用者、住西班牙且說西班牙語的使用者與住葡萄牙但說西班牙語的使用者等等。

資源以在 JSR 238 中定義的格式儲存在檔案中，資源檔案隨附於 MIDlet 套裝軟體 JAR 檔案。Sun Java™ Wireless Toolkit for CLDC 提供資源管理員，可簡化建立與維護資源檔案的工作。

---

### 17.1 設定模擬器的語言環境

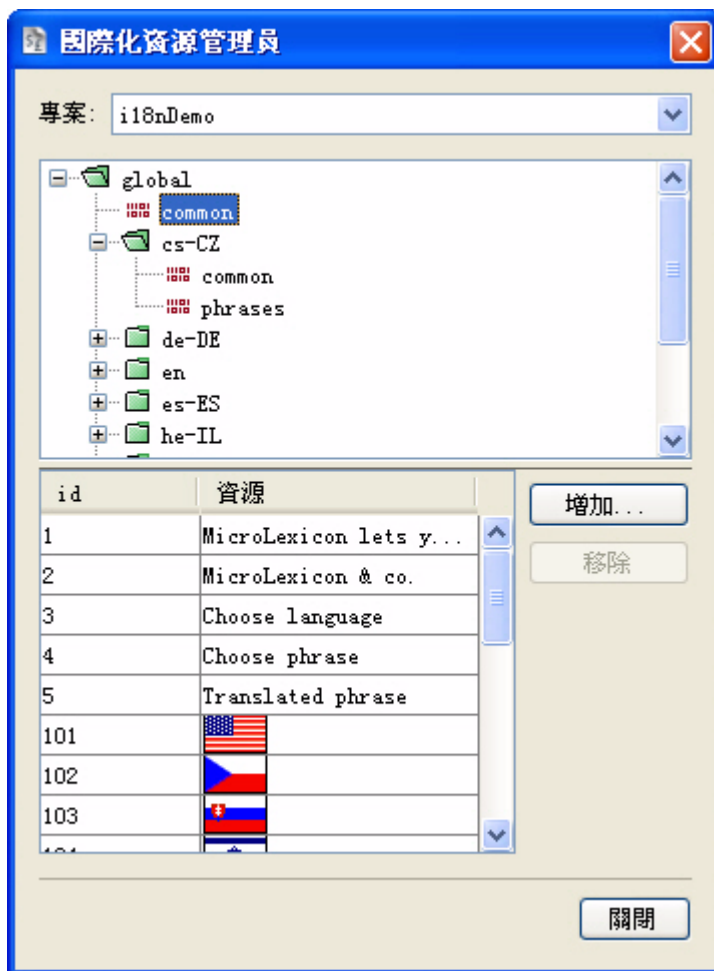
裝置的語言環境包含在系統特性 `microedition.locale` 中。您可以選擇 [編輯] > [偏好]，然後選取 [國際化] 來變更模擬器的語言環境。請從組合方塊中選擇一種語言環境或直接鍵入該語言環境。

---

### 17.2 檢視應用程式資源

若要啟動資源管理員，請選擇 [檔案] > [公用程式]。選取 [國際化資源管理員]，然後按一下 [啟動]。

圖 17-1 資源管理員



首先，請從【專案】功能表中選擇您的專案。所有用於已選取之專案的資源均將顯示在視窗其他部分。如果這是您首次檢視資源管理員，請檢視用於 i18nDemo 示範應用程式的資源，其中包括許多有趣的資源。

您可以按一下【+】號展開目錄或按一下【-】號將其摺疊。

幾乎所有作業均可透過在目錄或資源檔案上按一下滑鼠右鍵來達成。

視窗的頂部窗格顯示應用程式中資源檔案的階層，而底部窗格則顯示資源檔案的內容。在圖 17-1 中，將顯示最高層級之 common 資源檔案的內容。

---

## 17.3 使用語言環境

語言環境即最高層級 `global` 目錄下的目錄。語言環境目錄包括資源檔案，資源檔案則儲存應用程式使用的實際資源。

語言環境以 MIDP 2.0 規格中所描述的標準語言與國家或地區代碼表示。例如，`pt-BR` 表示居住在巴西說葡萄牙語的使用者。

若要增加語言環境目錄，請在最高層級之 `global` 目錄上按一下滑鼠右鍵，然後選擇 [ 增加語言環境 ]。請從組合方塊中選擇一種語言環境或直接鍵入該語言環境，然後按一下 [ 確定 ]。

若要重新命名一個語言環境，請在語言環境目錄上按一下滑鼠右鍵，然後選擇 [ 重新命名 ]。

若要移除某個語言環境及其包括的所有資源檔案，請在該語言環境目錄上按一下滑鼠右鍵，然後選擇 [ 刪除 ]。

---

## 17.4 使用資源檔案

資源檔案可以是全域的（在最高層級）也可以在某語言環境目錄下。若要建立新的全域資源檔案，請在最高層級之 `global` 目錄上按一下滑鼠右鍵，然後選擇 [ 增加新資源檔案 ]。為該檔案選擇一個名稱。

在資源檔案上按一下滑鼠右鍵，然後選擇 [ 重新命名 ] 以重新命名該檔案。

在資源檔案上按一下滑鼠右鍵，然後選擇 [ 刪除 ] 以移除該檔案。

您可以複製、剪下與貼上全部資源檔案。在某檔案上按一下滑鼠右鍵，然後選擇 [ 複製 ] 或 [ 剪下 ]。然後在語言環境目錄（或最高層級 `global`）上按一下滑鼠右鍵並選擇 [ 貼上 ]。

---

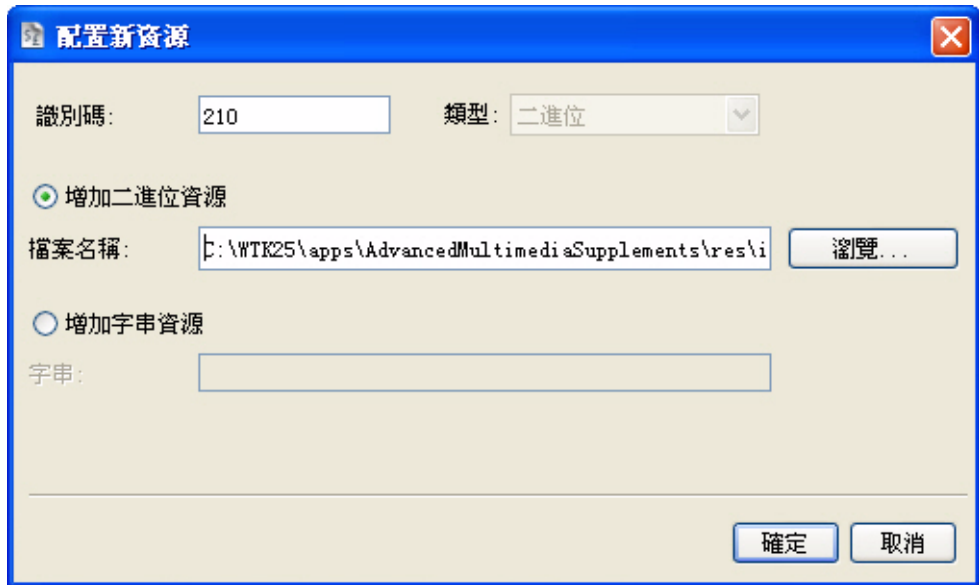
## 17.5 使用資源

在資源管理員視窗的頂部窗格按一下某資源檔案，以在底部窗格查看其內容。

若要增加新文字資源，請按一下 [ 增加 ]，然後選取 [ 增加字串資源 ]。系統將自動指定識別碼，但您要的話也可以輸入其他的識別碼。按一下 [ 確定 ]。

若要增加影像或其他類型的資料，請按一下 [ 增加 ]，然後選取 [ 增加二進位資源 ]。您可以瀏覽至您希望增加的所有檔案。而且如前所述，您要的話還可以變更該識別碼。按一下 [ 確定 ] 以增加該資源。

圖 17-2 增加影像資源



若要編輯某資源，請連按兩下該資源。您可以為字串資源鍵入新值。如果您連按兩下檔案資源，可以選擇其他檔案。

# 應用程式示範

本附錄描述 Sun Java™ Wireless Toolkit for CLDC 所附帶的應用程式示範。

## A.1 簡介

Sun Java™ Wireless Toolkit for CLDC 包含一些示範應用程式，以特別介紹模擬器所支援的一些技術與 API。這些示範的目的是要讓您大致瞭解一下模擬器的一些 API 功能與整個工具組的增強功能。

表格 A-1 列出此發行版本包含的所有示範應用程式。

大部分示範應用程式的執行方式都很簡單。第 A-3 頁的第 A.2 節「一般指示」包含執行大部分示範的說明。表格 A-1 中列出了包含其他文件的示範的連結。如果沒有連結，代表這個示範很簡單 ( 或有自己的說明 )，一般說明已足夠。

每個示範應用程式的原始碼位於 `toolkit/apps` 目錄中。子目錄內包含專案，而且每個專案各有一個內含 Java 程式設計語言原始碼的 `src` 目錄。例如，在 Windows 中，如果工具組安裝在 `C:\WTK2.5.2` 中，則 `WMA Demo` 中 SMS 寄件者 MIDlet (`example.sms.SMSSend`) 的原始碼就位於 `C:\WTK2.5.2\apps\WMA Demo\src\example\sms\SMSSend.java` 中。如第 1-2 頁的第 1.1.2 節「工作目錄檔案」中所討論，當您開啓某個專案時會將其複製到您的 `workdir/apps` 目錄。

表格 A-1 應用程式示範

示範	API	描述	特殊指示
AdvancedMultimedia Supplements	JSR 234 Advanced Multimedia Supplements	展示 3D 音效、迴聲、影像處理和相機控制。	A.3
Audiodemo	MMAPI 1.1	示範音效功能，包含混音以及配合動畫一起播放音效。	
BluetoothDemo	JSR 82 Bluetooth	示範使用藍芽探索裝置與交換資料。	A.4

**表格 A-1** 應用程式示範 (繼續)

示範	API	描述	特殊指示
CHAPIDemo	JSR 211 CHAPI	也利用 <code>MediaHandler</code> 的內容檢視器。	<a href="#">A.5</a>
CityGuide	JSR 179 Location API	展示根據目前位置顯示地標的市區地圖。	<a href="#">A.6</a>
Demos	MIDP 2.0	包含各種不同範例：動畫、色彩、網路、財務及其他。	<a href="#">A.7</a>
Demo3D	JSR 184 Mobile 3D Graphics	包含示範如何使用立即模式與保留模式兩種 3D 圖形的 MIDlet。	<a href="#">A.8</a>
FPDemo	CLDC 1.1	簡單的浮點計算機。	
Games	MIDP 2.0	包含 <code>TilePuzzle</code> 、 <code>WormGame</code> 和 <code>PushPuzzle</code> 。	
GoSIP	JSR 180 SIP	示範使用 SIP 和 SIP 伺服器啟動聊天功能	<a href="#">A.9</a>
i18nDemo	JSR 238 Mobile Internationalization API	包含字串排序、數字格式化和片語翻譯機。	<a href="#">A.10</a>
JBricks	JSR 229 Payment API	使用 <code>Payment API</code> 購買額外生命或等級的遊戲。	<a href="#">A.11</a>
JSR172Demo	Web Services	示範如何使用 JSR 172 API 從 MIDlet 連接到 Web 服務。	<a href="#">A.12</a>
MobileMediaAPI	MMAPI 1.1	示範 MMAPI 功能，包含鈴聲序列、MIDI 播放、取樣的音效播放與視訊。	<a href="#">A.13</a>
NetworkDemo	MIDP 2.0	展示如何使用資料封包與串列連線。	<a href="#">A.14</a>
ObexDemo	JSR 82 OBEX	示範透過 IrDA 使用 OBEX 傳輸資料。	<a href="#">A.15</a>
OpenGL® ES Demo	JSR 239 OpenGL® ES	展示如何使用 OpenGL® ES 建立 3D 圖形。	
PDAPDemo	JSR 75 PIM 和 FileConnection	示範如何處理連絡人、行事曆項目和待辦事項。示範存取本機檔案。	<a href="#">A.16</a>
Photoalbum	MIDP 2.0	示範各種不同影像格式。	
SATSA Demos	JSR 177 SATSA	示範與智慧卡和 SATSA 的其他功能通訊。	<a href="#">A.17</a>
SATSAJCRMIDemo	JSR 177 SATSA	展示如何使用 SATSA-Java Card RMI。	<a href="#">A.18</a>



表格 A-1 應用程式示範 (繼續)

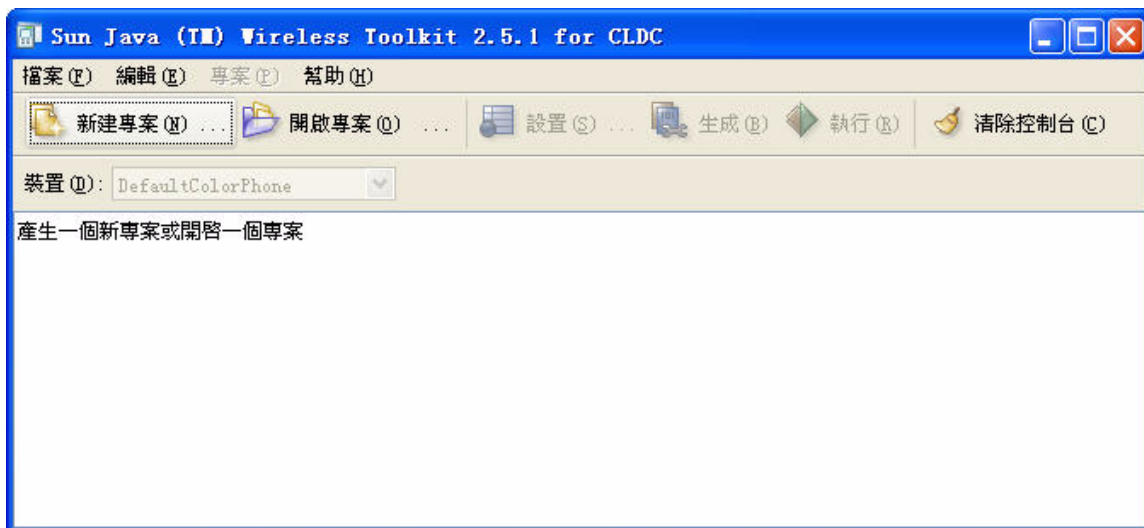
示範	API	描述	特殊指示
SIPDemo	JSR 180 SIP	使用 SIP 進行簡單訊息交換。	<a href="#">A.19</a>
SnapMobileSample		示範連線多人遊戲功能。	
SVGContactList	JSR 226 SVG API	示範以不同的面板顯示連絡人清單。	<a href="#">A.20</a>
SVGDemo	JSR 226 SVG API	展示描繪 SVG 的不同技術。	<a href="#">A.21</a>
UIDemo	MIDP 2.0	展示 MIDP 2.0 使用者介面功能的廣度	
WMADemo	WMA 2.0	展示如何傳送及接收 SMS、CBS 與 MMS 訊息。	<a href="#">A.22</a>

## A.2 一般指示

大部分示範應用程式可在執行後無需特殊準備便可啟動。但是，某些示範需要變更工具組偏好或設定。這一節描述一般程序。

第一個步驟是執行工具組。若要執行此動作，請移至 Microsoft Windows 的 [開始] 功能表，選擇 [開始] > [所有程式] > [Sun Java Wireless Toolkit 2.5.2 for CLDC] > [Wireless Toolkit 2.5.2]。將顯示使用者介面，如圖 A-1 中所示。

圖 A-1 Wireless Toolkit 使用者介面



按一下 [ 開啓專案 ] 按鈕，開啓示範應用程式。會顯示所有可用應用程式的清單。如第 1-2 頁的第 1.1.2 節「工作目錄檔案」中所討論，安裝目錄中的專案為斜體，而您工作目錄中的專案以粗體顯示 (圖 1-1)。當您開啓安裝目錄中的某個專案時，會在您的工作目錄中建立一個副本，然後開啓。

開啓應用程式後，您可以按工具列上的 [ 執行 ] 按鈕；或者如果需要在模擬器上安裝，則請按一下 [ 專案 ] > [ 經由 OTA 執行 ]。

開啓裝置模擬器視窗，顯示執行中的示範應用程式。如果有 MIDlet 的功能表，請使用導覽箭頭選擇一個項目，然後選擇 [ 選取 ]。示範進行時，您可能需要按下左側或右側螢幕下的軟鍵之一。使用軟鍵可以安裝或啟動應用程式、開啓功能表、結束或執行某些其他動作。某些範例包含這些指令。

有些示範需要特定設定與指示。例如，如果某個範例使用 Web 服務並且您在防火牆後方，則您必須配置模擬器的代理伺服器設定，否則示範將失敗：

- 選擇 [ 編輯 ] > [ 偏好 ]
- 選取 [ 網路配置 ] 圖示。
- 核取 [ 使用代理伺服器 ]。
- 填入代理伺服器位址欄位以及連接埠號。

如需更多操作指示，請閱讀每個示範的說明。

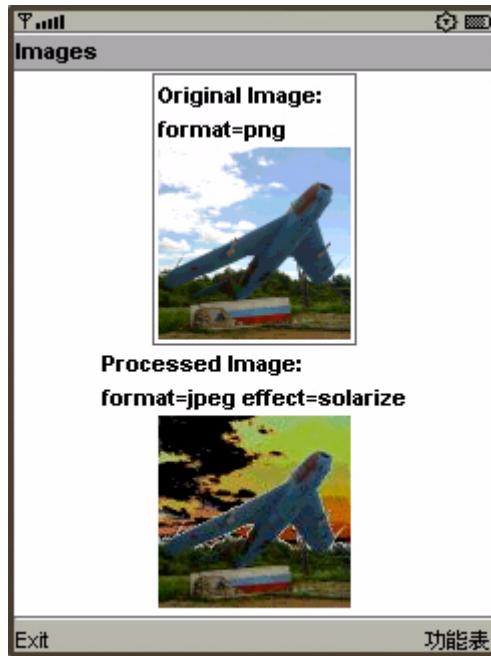
---

## A.3 Advanced Multimedia Supplements

此 MIDlet 套裝軟體示範了 JSR 234 Advanced Multimedia Supplements (AMMS) 的功能。它包含以下 MIDlet：

- **影像效果** — 展示標準影像處理作業。從功能表中選擇一種效果。處理過的影像顯示在來源影像後面。請注意，某些項目（例如設定轉換）可以執行數項作業。按一下 [ 完成 ] 軟式按鈕以套用每種效果。

圖 A-2 在 MIDlet 中處理影像



- 收音機選台器 — 透過工具組的內建 HTTP 伺服器從專案目錄讀取音效檔案，可模擬收音機選台器。
- 相機 — 展示 Advanced Multimedia Supplements 如何控制裝置的相機。此螢幕顯示相機的取景器（使用動態影像模擬）。您可以使用功能表中的指令變更相機設定，以及拍攝並管理快照。
- 移動直升機 — 模擬直升機在固定觀察者周圍飛行的場景。請使用耳機，以取得最佳效果。您可以控制許多模擬參數（包括直升機的大小）、是否使用都卜勒效果，以及觀賞者的方位（例如是直立還是趴著）。
- 音樂效果 — 展示 Advanced Multimedia Supplements 的進階音效功能。音效檔案連續循環播放時，您可以調整音量、聲像、等化器設定、迴聲和合聲設定。

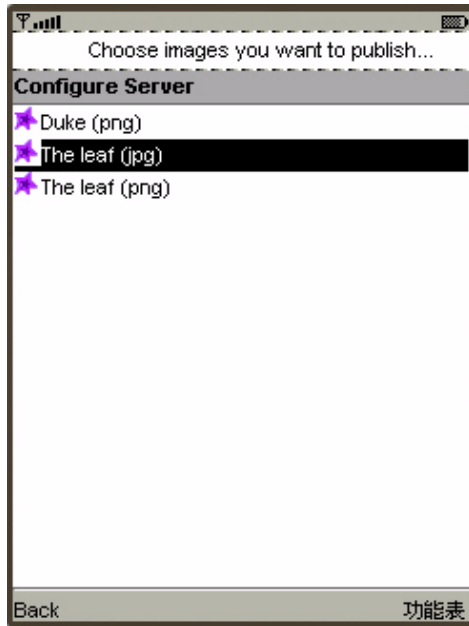
## A.4 藍芽示範

此應用程式包含示範如何使用 JSR 82 藍芽 API 的 MIDlet。

BluetoothDemo 專案展示如何在兩個使用藍芽的裝置之間傳輸影像。您必須執行兩個模擬器實例，才能瞭解此示範的運作方式。

在第一個模擬器中，啟動 [ 藍芽示範 ]，然後選擇 [ 伺服器 ]。模擬器會詢問您是否允許藍芽連線。選擇「是」。伺服器會啟動，並顯示影像清單。一開始，藍芽網路上沒有任何可用的影像。若要使影像可用，請選取這些影像，然後從功能表選擇 [ 發佈影像 ]（或鍵入或按一下 1）。圖示顏色會從紫色變為綠色，表示已經發佈。

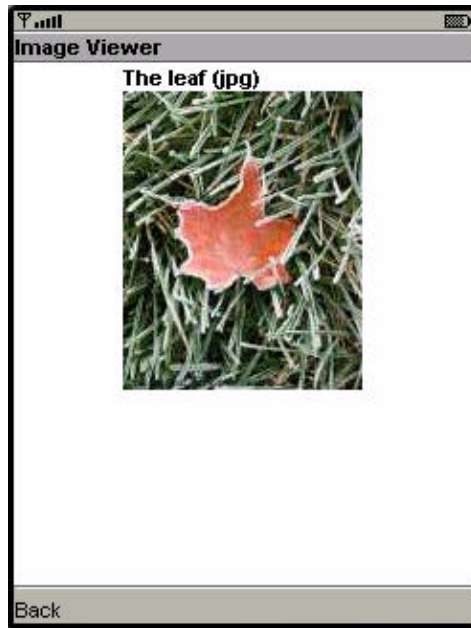
■ A-3 執行藍芽示範伺服器



在第二個模擬器上，啟動 [ 藍芽示範 ]，然後選取 [ 用戶端 ]。MIDlet 會告訴您，已準備好搜尋影像。按一下 [ 尋找 ] 軟式按鈕。MIDlet 會尋找其他模擬器，並從中取得影像清單。從清單中選取一個影像，並選擇 [ 載入 ]。模擬器會詢問您是否允許連線。選擇「是」。

- 如果您正於信任的保護網域中執行示範，則影像會使用模擬的藍芽傳輸，並且會顯示在用戶端模擬器上。
- 如果您未於信任的保護網域中執行，則第一個模擬器（伺服器）會顯示一則提示，詢問您是否要授權來自該用戶端的連線。選擇「是」。影像會顯示於用戶端模擬器中。

圖 A-4 使用模擬藍芽傳輸的影像



---

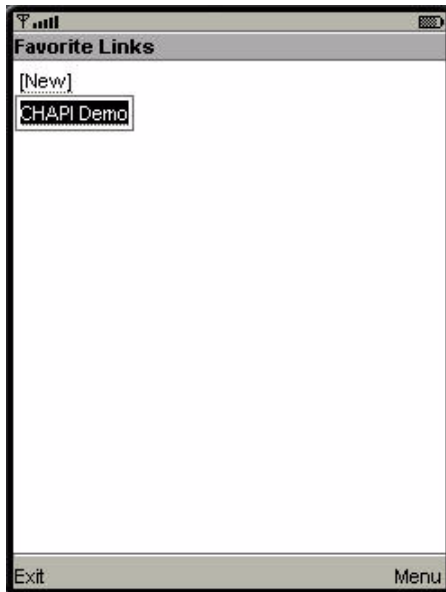
## A.5 CHAPIDemo

CHAPIDemo 是一個內容瀏覽器 (請參閱 JSR 211)，裡面保留一個我的最愛清單，並可讓您選取與檢視各種內容。

本示範使用內容處理器註冊表，因此您使用【執行】按鈕時無法看到其所有功能。請改用【專案】>【經由 OTA 執行】將應用程式安裝至模擬器。如果不知道如何執行，請查閱「[第 2-8 頁的第 2.3.2 節「安裝」](#)」。

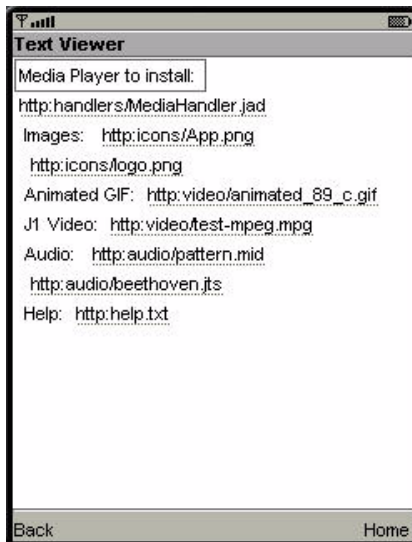
安裝完 CHAPIDemo 後，會在【文字檢視器】中顯示為應用程式清單。它是純文字的 MIDlet 內容處理器。選取【文字檢視器】，然後從軟式按鈕功能表中選擇【啟動】。會顯示我的最愛連結清單。

圖 A-5 檢視 CHAPIDemo 中的我的最愛連結



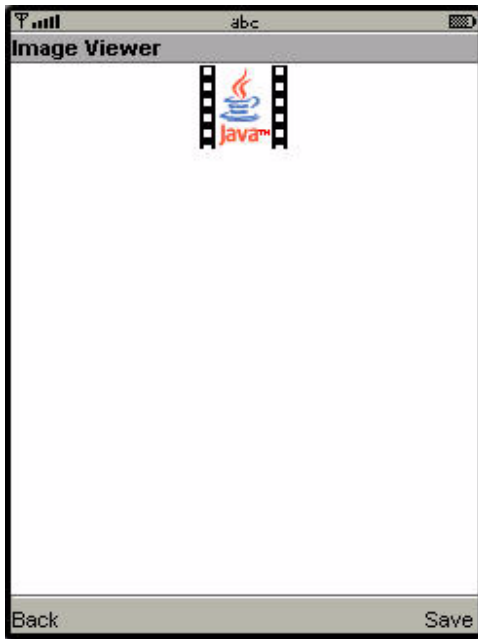
使用導覽鍵高亮度顯示 CHAPIDemo，然後按下模擬器上的 [ 選取 ]。應用程式會詢問是否可以使用通話時間。按下 [ 是 ] 軟式按鈕。會顯示各種類型內容的清單 ( 圖 A-6 )。

圖 A-6 內容清單



嘗試選取一個 Duke.png。使用箭頭高亮度顯示該連結，然後按下 [ 選取 ] 以檢視此檔案。使用 CHAPI 啟動 ImageViewer MIDlet，並會顯示內容。

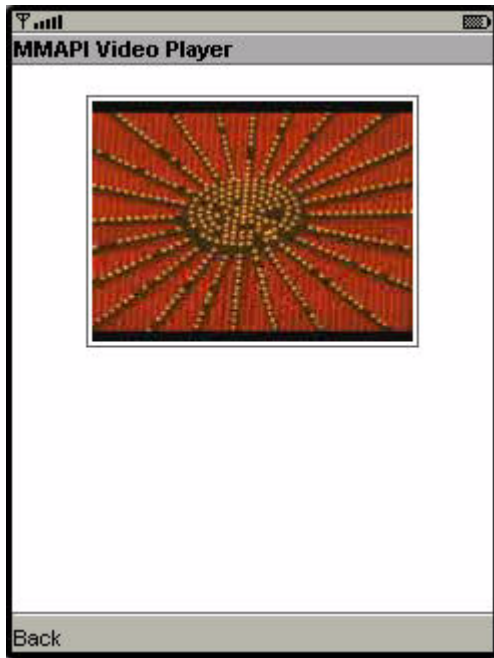
圖 A-7 檢視 PNG 影像



其他類型的內容需要另一個內容處理器 MIDlet 套裝軟體：MediaHandler。若要從 CHAPIDemo 安裝此套裝軟體，請選取 MediaHandler.jad 連結 (圖 A-6 中所示清單中的第一個項目)。會呼叫 AMS，並引導您完成整個安裝。

安裝完 MIDlet 套裝軟體之後，您可以檢視 [文字檢視器] 中列出的其他類型內容。例如，選取 <http://video/test-mpeg.mpg> 以查看一系列影像 (包括圖 A-8 中顯示的影像)。

圖 A-8 使用 MediaHandler 檢視 MPEG 電影



若要檢視 `TextViewer` 和 `ImageViewer` MIDlet 的內容處理器設定，請按一下 [ 設定 ]，然後按一下 [ 內容處理器 ] 圖示。您可能還希望檢查 `MediaHandler` 專案。

## A.6 CityGuide

`CityGuide` 會示範如何使用位置 API (JSR 179)。它可在市區地圖上顯示行人的目前位置。行人會在市區內移動，行人接近地標時，地標也會高亮度顯示並註明。在此示範中，我們透過名稱爲 `citywalk.xml` (事件檔案) 的 XML 程序檔取得行人的位置，該程序檔可提交裝置位置資訊。請參閱第 13 章以獲得完整說明。

由於位置提示會經常出現，因此最佳方法是以製造商 (可信任) 模式執行此示範，如第 6-3 頁的第 6.2.1 節「MSA 保護網域」中的解釋。在使用者介面中，選取 [ 編輯 ] > [ 偏好 ]，然後選取 [ 安全 ]。爲安全網域選擇 [ 製造商 ]。

開啓並執行 `CityGuide` 專案。在模擬器中，啓動 `CityGuide` MIDlet。按 [ 下一步 ] 以檢視地圖頁面。



圖 A-9 您在城市中的位置



從模擬器視窗功能表中選擇 [MIDlet] > [外部事件]。在 [位置] 標籤上，按一下 [瀏覽] 按鈕。選取 `workdir\apps\CityGuide\citywalk.xml` 中的以下事件檔案。

視窗底部的播放器按鈕現在正在使用中。請參閱圖 13-1。按下綠色播放按鈕（向右指的三角形）以執行程序檔。

顯示螢幕出現四種類型的地標：餐廳、博物館、商店和電影院。若要調整地標顯示，請開啓軟式功能表並選擇 [設定] 指令。請參閱圖 A-10。使用導覽鍵高亮度顯示某個種類，然後使用 [選取] 核取或取消核取某個項目。

當您在某個地標（如地圖上高亮度顯示的地標所示）附近時，請開啓軟式功能表並選擇 [詳細資訊] 指令，以查看更多資訊。如需有關位置程序檔的更多資訊，請參閱第 13 章。

圖 A-10 位置設定

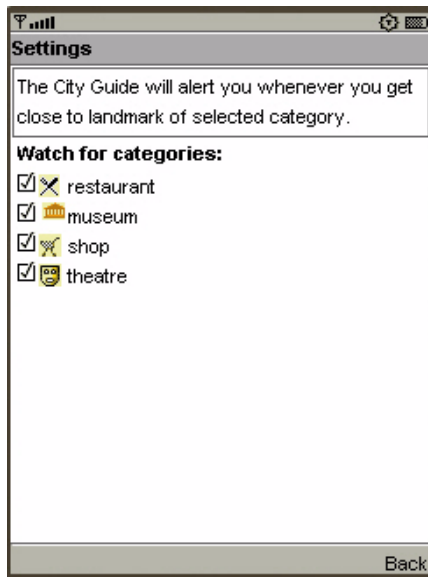
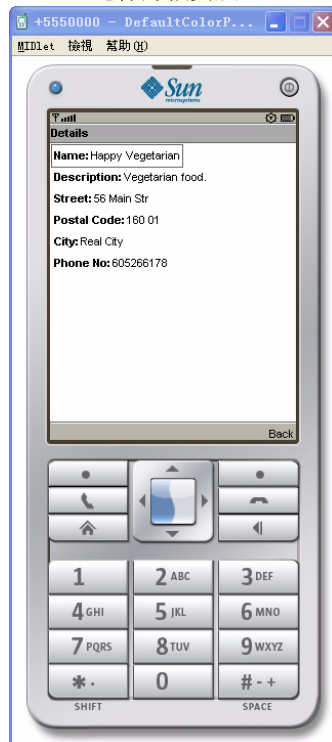


圖 A-11 地標詳細資訊



## A.7 示範

本示範包含幾個 MIDlet，強調不同的 MIDP 功能。

### A.7.1 色彩

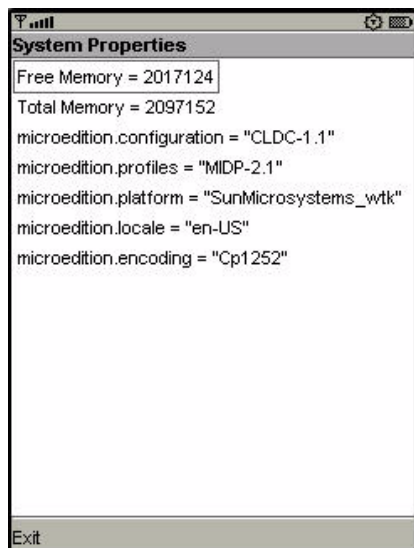
此應用程式會顯示一個大塊橫向矩形，跨越整個螢幕寬度。下面有十個小塊縱向矩形，橫跨整個螢幕。最後有三條水平色棒，分別顯示藍、綠和紅色 (RGB) 的值。系統會根據第一個功能表的選取，以十進位 (0-255) 或十六進制 (00-ff) 數字表示值。

- 若要選取垂直色棒以進行變更，請使用向上導覽箭頭移動到色棒。使用向右導覽箭頭高亮度顯示色棒。大塊矩形會變成所選色棒的顏色。
- 使用向上或向下選取箭頭選擇要變更的值 (紅、綠或藍)。使用向左或向右箭頭增加或減小所選值。第二個功能表項目可讓您以 4 (微調) 或 32 (粗調) 為增減單位。
- 您可以變更任意或所有垂直色棒上的顏色。

### A.7.2 特性

此 MIDlet 會顯示特性值。例如，請參考圖 A-12：

圖 A-12 系統特性



## A.7.3 Http

此測試應用程式使用 HTTP 連線請求網頁。請求是透過 HTTP 協定 GET 或 POST 方法發佈的。如果已使用 HEAD 方法，則會從請求讀取 HEAD 特性。

### 準備執行示範

開始之前，請按以下程序檢查您的設定。

- 移至 [ 偏好 ] > [ 安全 ]。將策略設定為 JTWI，並將網域設定為最大值。
- 在 [ 偏好 ] > [ 網路配置 ] 中，HTTP 版本必須為 1.1。
- 如果您在防火牆後方，請移至 [ 偏好 ] > [ 網路配置 ]，並指定代理伺服器資訊。
- 如果您正在執行防毒軟體，則可能需要建立允許此 MIDlet 的規則，以允許連線至特定的網站或從特定的網站進行連線。

### 執行示範

啟動 Http MIDlet。依預設，MIDlet 嘗試聯絡 `http://www.yahoo.com`。若要進行測試，請選擇 [ 功能表 ] 軟鍵並選擇 1、2 或 3 測試所選 URL。

[Http 測試] 會傳回自己能取得的資訊。如果資訊填滿螢幕，請使用向下箭頭捲動到結尾。資訊量取決於請求類型和此頁面提供的 META 資訊量。若要提供內文資訊或內容，此頁面必須依 RFC 2616 中的描述宣告 CONTENT-LENGTH。

### 使用功能表選項

使用 [ 功能表 ] 軟鍵進行以下動作。

- 選擇 1 從所選頁面中 GET ( 獲得 ) 資訊。
- 選擇 2 從所選頁面中獲得 POST 資訊。
- 選擇 3 顯示此頁面的 HEAD 屬性。
- 選擇 4 開啓目前的網頁清單。您可以選擇新頁面，或將您自己的頁面增加到清單中。若要指定新的 URL，請選擇 [ 功能表 ] 軟鍵並選擇 4。螢幕會顯示 [http://]。鍵入 URL 的其餘部分，確定以斜線 (/) 結尾。例如 `http://www.internetnews.com/`。按下 [ 確定 ] 軟式按鈕。[Http 測試] 螢幕會顯示新的 URL 以及動作的提示。

## A.7.4 FontTestlet

此 MIDlet 展示各種可用字型：Proportional、Regular、Regular Italic、Bold Plain 和 Bold Italic。從功能表中選擇 1 或 2，以在系統字型 ( 細明體 ) 和固定間距字型間切換。

## A.7.5 股票

正如 **Http** 示範，此範例使用 HTTP 連線獲得資訊。使用與第 A-14 頁的第 A.7.3 節「**Http**」相同的準備步驟。

執行示範專案並啟動 **Stock MIDlet**。

依預設，畫面會在頂部顯示空的快訊列。快訊下方的功能表清單會顯示四個應用程式：「**Stock Tracker (股票追蹤器)**」、「**What If?**」、「**Alerts (警報)**」和「**Settings (設定)**」。使用前三個應用程式之前，您必須增加股票代號。

### A.7.5.1 使用設定

若要使用此應用程式的功能，您必須為應用程式提供一些股票代號才能使其運作。

#### 將股票代號增加到快訊

若要將股票代號增加到快訊，請使用導覽箭頭選取 [ 設定 ]。

選取 [ 新增股票 ]。

螢幕會提示您輸入股票代號。鍵入 **SUNW** 並選取 [ 完成 ] 軟鍵。您增加的股票和目前價格會顯示在快訊中。增加更多代號，例如 **IBM** 和 **HPQ**。

#### 變更更新間隔

依預設，更新間隔為 15 分鐘。選取 [ 更新 ] 以變更時間間隔。使用導覽箭頭選取 [ 連續 ]、[ 15 分鐘 ]、[ 30 分鐘 ]、[ 1 小時 ] 或 [ 3 小時 ] 中的一個選項。選取 [ 完成 ] 軟鍵。

#### 移除股票

選取 [ 移除股票 ]。可以看到您已增加的股票清單。使用導覽鍵選取一個或多個要移除的股票。選擇 [ 完成 ] 軟鍵。

### A.7.5.2 股票追蹤器

股票追蹤器可顯示您已增加的股票及其現值清單。股票追蹤器可顯示有關已選取股票的其他資訊，例如，最後成交價和最高價與最低價。

選擇一種股票並按下 [ 選取 ]。

### A.7.5.3 What If?

What If? 是一個應用程式，會詢問您擁有股票的原始購買價格和股數。它可根據目前價格計算損益。

選取股票代號。

輸入購買價格和股份，然後按下 [ 計算 ]。

### A.7.5.4 警報

當價格變更為您指定的值時，此應用程式會傳送給您一則通知。

從主功能表中選取 [ 警報 ]。

選取 [ 增加 ]。

選取 [ 股票 ]。螢幕會顯示提示 [ 股票到達以下價格時發出警報 ]。輸入一個整數。

警報位於 [ 目前的警報 ] 清單中。若要移除某個警報，請按下 [ 移除 ] 並選取該警報。選擇 [ 完成 ] 軟鍵。

達到您指定的價格時，您會聽到鈴聲並收到一則訊息。例如，*Symbol* 已達到您的價格點 *\$value* 並且目前正以 *\$current\_value* 價格成交。此警報觸發之後，便會從 [ 目前的警報 ] 清單中消失。

## A.7.6 購票

示範線上購票拍賣應用程式的可能運作方式。主螢幕會在頂部顯示購票快訊。依預設，[ 選擇樂團 ] 欄位會顯示 [ 艾拉妮絲莫莉塞特 ]。

若要選擇樂團，反白顯示樂團名稱並按下 [ 選取 ]。使用向下箭頭反白顯示不同的樂團 (例如「魔比」)，然後按下 [ 選取 ]。會顯示可用的拍賣。

若要投標，請選取 [ 功能表 ] 軟鍵並選擇 [2]。使用方向鍵在欄位間移動。填妥每個欄位。選取 [ 下一步 ] 軟鍵。應用程式會要求您確認您的投標。使用方向鍵反白顯示 [ 提交 ]，然後按下 [ 選取 ]。您會收到一個確認碼。按一下 [ 樂團 ] 以返回到歡迎頁面。

若要設定警報，請選取 [ 功能表 ] 軟鍵並選擇 3。使用導覽箭頭移動到此欄位並鍵入一個高於目前投標價的值。選取 [ 儲存 ] 軟鍵。會返回到歡迎頁面。只要以超過警報值的價格投標，就會觸發警報。您的設定決定了應用程式檢查變更的頻率，因此可能要過幾分鐘才能聽到警報。

若要加入樂團，請選取 [ 功能表 ] 軟鍵並選擇 4。鍵入樂團名稱或用逗號隔開的名稱清單。選擇 [ 儲存 ] 軟鍵。確認後便會返回到歡迎頁面。增加的樂團顯示在 [ 選擇樂團 ] 下拉式清單中。

---

**備註** – 這只是個示範。若要完整描述樂團，您必須編輯檔案 `workdir\apps\Demos\src\example\auction\NewTicketAuction.java`。

---

若要移除樂團，請選取 [ 功能表 ] 軟鍵並選擇 5。導覽至某個樂團，然後選擇 [ 選取 ] 以標記核取方塊。您可以選擇多個樂團。選擇 [ 儲存 ] 軟鍵。

若要顯示快訊顯示、更新、警報音量和日期的目前設定，請選取 [ 功能表 ] 軟鍵並選擇 6。如果需要，請使用方向鍵和選取鍵變更這些值。選擇 [ 儲存 ] 軟鍵。

## A.7.7 ManyBalls

此 MIDlet 一開始是一個球在螢幕中移動。使用向上、向下箭頭提高或降低球速 (fps)。使用向左、向右鍵增加或減少球數。

---

## A.8 Demo3D

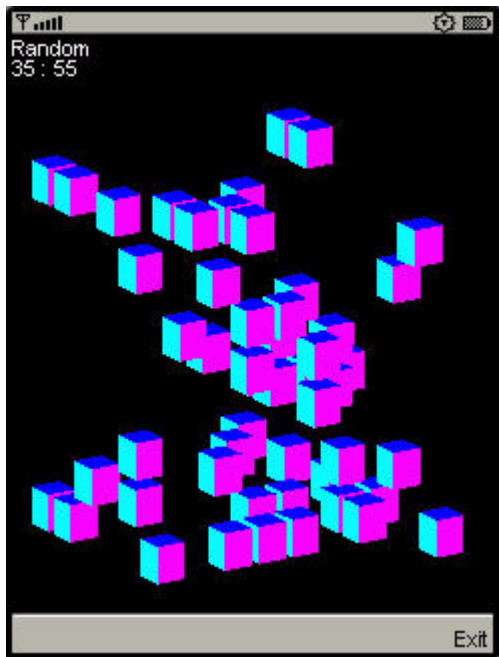
此應用程式包含三個 MIDlet，以顯示模擬器的 JSR 184 (Mobile 3D Graphics API) 支援。

### A.8.1 Life3D

Life3D 以三度空間實作流行的生命之旅 (Game of Life)。活的細胞以立方體表示。每個細胞可能有 26 個鄰居 (包含對角)。在動畫的每個步驟中，少於四個鄰居的細胞會因為太寂寞而死掉，而多於五個鄰居的細胞則因為太擁擠而死掉。剛好四個鄰居的空細胞將會變成新的活細胞。

遊戲板的圖形會慢慢旋轉，讓您可以從各種角度看到遊戲板。

圖 A-13 三度空間的生命之旅



使用表格 A-2 中的小鍵盤按鈕控制遊戲。

表格 A-2 對 Life3D 的控制

按鈕	描述
0	暫停動畫。
1	以預設速度重新繼續動畫。
2	加快動畫的速度。
3	放慢動畫的速度。
4	從任意清單中選擇上一個預設配置。配置的名稱顯示在螢幕頂端。
5	選擇清單中的下一個預設配置。
*	產生隨機配置，並進行動畫模擬，直到達到穩定或死亡。如果死亡，便會產生新的隨機配置。

本範例的原始碼有非常完善的說明記錄。請參閱 `toolkit\apps\Demo3D\src\com\superscape\m3g\wtksamples\life3d\Life3D.java`。



## A.8.2 PogoRoo

PogoRoo 展示一隻袋鼠用彈簧單高蹺上、下彈跳。若要掌控袋鼠，請使用方向鍵。向上可以往前跳，向下往後跳，向左、向右則可以改變方向。您必須按住按鍵才能看到效果。

圖 A-14 彈跳中的袋鼠



## A.8.3 retainedmode

retainedmode MIDlet 會播放一個場景的檔案，顯示一個永不疲倦的滑板選手，無止盡地繞圈圈。

圖 A-15 永不疲倦的滑板選手



---

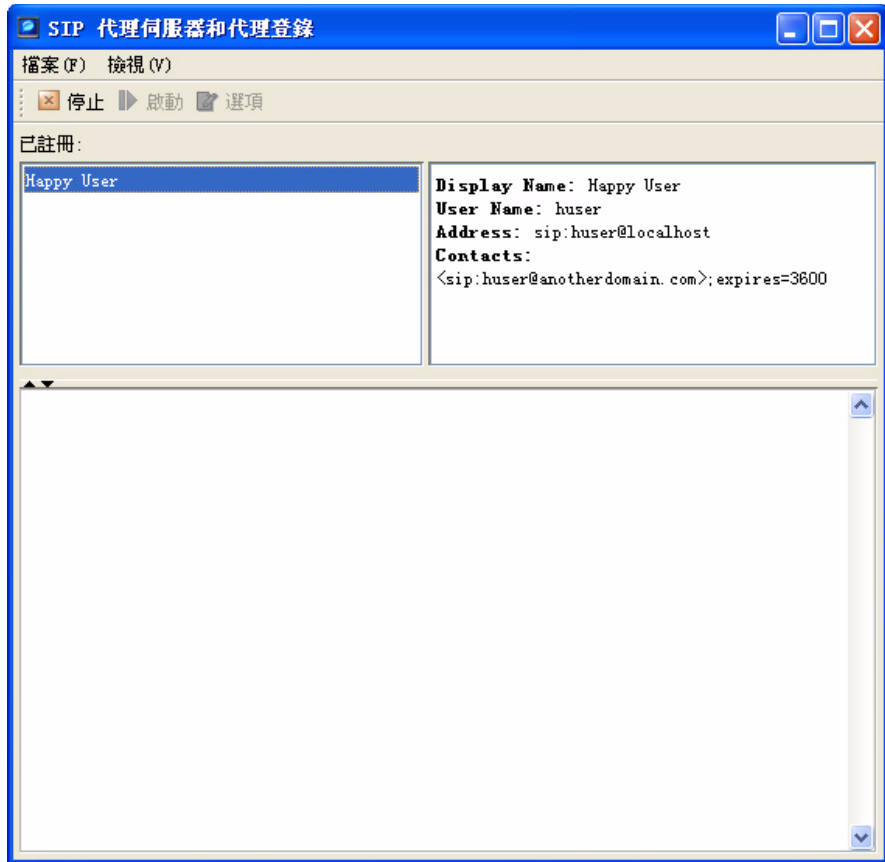
## A.9 GoSIP

GoSIP 是一個聊天應用程式，以 SIP (JSR 180) 設定使用 SIP 代理伺服器和代理登錄的通訊。

由執行 SIP 伺服器開始。選擇 [ 檔案 ] > [ 公用程式 ]。選取 [ 啟動 SIP 伺服器 ]，然後按下 [ 啟動 ]。會顯示 SIP 代理伺服器視窗。按一下 [ 啟動 ] 以執行此伺服器。

接著使用 GoSIP 應用程式執行兩個模擬器的範例。

圖 A-16 SIP 代理伺服器 and 代理登錄



在第一個模擬器中，啟動 Sippy A。系統提示您輸入代理主機時，請輸入您的本地機器名稱或 IP 位址並選擇 [ 下一步 ]，然後選擇 [ 註冊 ]。SIP 伺服器視窗會顯示模擬器的 SIP 訊息。在已註冊使用者的清單中會顯示 [Sippy A]。模擬器會建議您邀請您的朋友 Sippy B 交談。但現在還不要執行。

在第二個模擬器中，啟動 Sippy B。和之前一樣，輸入 SIP 代理伺服器的位址、選擇 [ 下一步 ]，然後選擇 [ 註冊 ]。[Sippy B] 使用者會顯示在 SIP 伺服器視窗中。

在第一個模擬器中選擇 [ 邀請 ]。第二個模擬器會指示其正在響鈴。選擇 [ 接聽 ] 開始聊天。兩個模擬器現在均顯示 [ 聊天中 ] 螢幕。您可以使用 [ 傳送 ] 指令來回傳送訊息。

完成後，選擇 [ 再見 ] 結束聊天。

## A.10 i18nDemo

此 MIDlet 套裝軟體展示 JSR 238 Mobile Internationalization API。MIDlets String Comparator 和 Formatter 展示如何排序字串以及針對不同的語言環境適當地顯示數字。第三個 MIDlet (MicroLexicon) 是一個小型片語翻譯程式，如果您需要在布拉格、赫茲利雅、北京、米蘭或幾個其他地方點杯啤酒，便可派上用場。

---

**備註** – Sun Java™ Wireless Toolkit for CLDC 的預設字型不支援中文和日文。若要使用這些語言，請在執行此示範之前執行以下步驟：

1. 安裝支援中文或日文的 True Type 字型。
  2. 修改 `toolkit\wtklib\devices\skin-directory\skin.properties` 以指定該字型。
- 

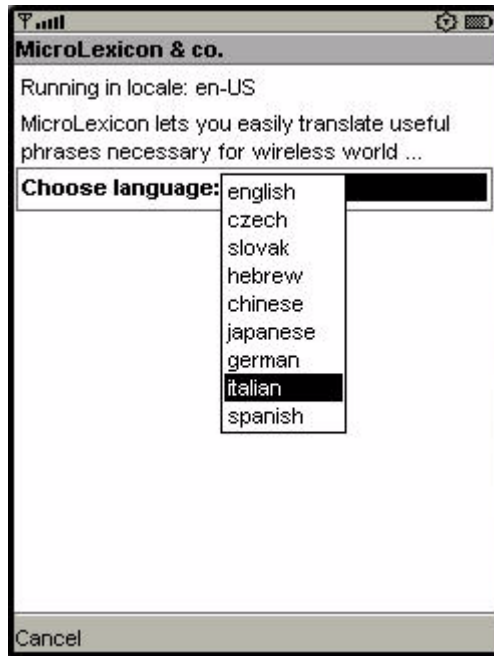
若要執行 MIDlet，請使用 [ 選取 ] 反白顯示此 MIDlet，然後使用右下按鈕啟動此 MIDlet。

String Comparator MIDlet 可示範每個語言環境的字串（城市名稱）排序之區別。啟動 MIDlet。使用右下按鈕檢視功能表。按一下或鍵入 2 以選取 [ 排序 - 預設 ]，清單即會按字母順序排序。按一下或鍵入 3 以選取 [ 排序 - 斯洛伐克語 ]。不管頂部有沒有標記，都很容易看出以字母 Z 開頭的城市差別。按一下 [ 結束 ] 以返回到 MIDlet 的清單。

第二個 MIDlet (Formatter) 只是針對不同語言環境的格式顯示時間和數字。按一下 [ 下一步 ] 以檢視所有四個螢幕。按一下 [ 結束 ] 以返回到 MIDlet 的清單。

最後一個 MIDlet (MicroLexicon) 將片語從一種語言翻譯為另一種語言。若要從清單中選取目標語言，請使用導覽箭頭反白顯示 [ 選擇語言 ]。按一下 [ 選取 ] 以檢視語言下拉清單。使用導覽箭頭選擇一種語言（請參考圖 A-17），然後按一下 [ 選取 ]。

圖 A-17 選擇目標語言



MicroLexicon 顯示片語的清單。反白顯示一個片語，然後按下模擬器上的 [ 選取 ] 按鈕。您可以看到目標語言的旗標和已翻譯的片語。

若要變更來源語言，請選擇 [ 編輯 ] > [ 偏好 ]。按一下 [ 國際化 ] 標籤，然後輸入有效的語言環境字串。下次執行模擬器和 MicroLexicon 時，如果支援，指示文字就會以指定的語言環境顯示。例如，cs-CZ 就可行。

MicroLexicon 採用 MIDlet 資源。若要瞭解如何使用工具組來本地化應用程式，請選擇 [ 專案 ] > [ 國際化資源管理員 ]。會顯示所有資源，包括文字和影像 ( 由 MicroLexicon 使用 )。您可以編輯資源，然後再次執行 MicroLexicon，看看有什麼變化。您無需再次生成應用程式，因為資源已經在執行階段載入。

圖 A-18 國際化資源管理員



這些資源本身儲存於 `workdir\apps\i18nDemo\res\global`。

## A.11 JBricks

JBricks 是一個遊戲，示範了 JSR 229 Payment API 的使用。此遊戲類似 Breakout 或 Arkanoid (打磚塊)。在 JBricks 中，您可以購買生命或新的遊戲等級。Payment API 在背景處理細節。

若要使用 JBricks 的支付款項功能，請使用 [ 專案 ] > [ 經由 OTA 執行 ] 將 JBricks 安裝至模擬器。如果不知道如何執行，請查閱「[第 2-8 頁的第 2.3.2 節「安裝」](#)」。

若要查看 JBricks 如何使用 Payment API，請從遊戲主功能表中選擇 [ 購買生命 ] 或 [ 購買等級 ]。接著選擇要購買一條生命，還是以優惠價格一次購買三條生命。下一個螢幕讓您選擇支付款項類型。

圖 A-19 選擇支付款項類型



使用導覽箭頭選取以 [ 透過以下方式支付 ] 開頭的行。按一下 [ 選取 ] 按鈕以查看下拉式功能表中可能的信用卡介面。使用導覽箭頭選取 [VISA 介面]，然後按一下 [ 選取 ]。按一下右下方的 [ 是 ] 以繼續進行。

然後，您便可以輸入信用卡資訊。使用任一有效的 VISA 號碼 (例如，4111111111111111) 和有效的過期日期。

圖 A-20 提供支付款項資訊



信用卡支付款項

金額 1.50 EUR

信用卡類型 VISA

持卡人

Jay Bricks

卡號 6111611161116111

過期日期 (mm/yyyy) 11/2011

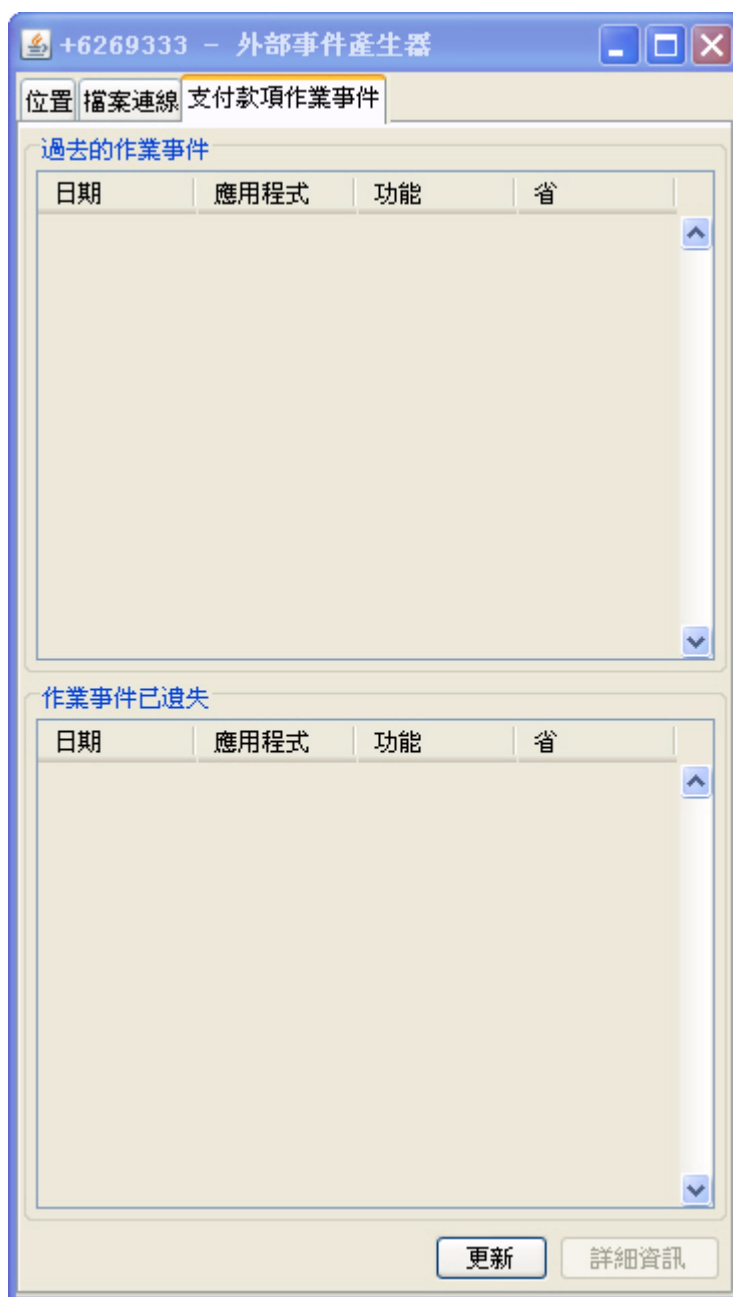
驗證 \*\*\*

取消 購買

若要檢視目前模擬器實例的作業事件，請選擇 [MIDlet] > [ 外部事件 ]，然後按一下 [ 支付款項作業事件 ] 標籤。會顯示此特定模擬器實例的作業事件。



圖 A-21 檢視作業事件



此外，您可以檢視通過工具組的支付款項系統的所有作業事件。選擇 [ 檔案 ] > [ 公用程式 ]，然後選取 [ 支付款項主控台 ]。主控台內的作業事件如下：

```
PSP Console running, using phone number +5550001.  
PSP Server running at https://localhost:-1  
Received Payment Request from 127.0.0.1  
  Credit card issued by: VISA  
  Credit Card type: 0  
  Credit Card Number: 4111111111111111  
  Credit Card Holder: Jonathan Knudsen  
  Feature ID: 3_lives  
  Credit Card Verification Number (CCV): 123  
  Payload: null  
Response to 127.0.0.1  
HTTP/1.1 200 OK  
Content-Length: 0  
Pay-Response: SUCCESSFUL  
Pay-Timestamp: 1156282954734
```

---

## A.12 JSR172Demo

JSR172Demo 展示如何從 MIDlet 存取 Web 服務。Web 服務已經在網際網路伺服器上執行。如果您在防火牆後方，您必須配置模擬器的代理伺服器設定。選擇 [ 編輯 ] > [ 偏好 ]，然後選取 [ 網路配置 ]。填入代理伺服器位址檔案以及連接埠號。生成並執行範例。

JSR172Demo 包含一個 MIDlet，名稱爲 Server Script。請啓動它，並依照提示執行。您可以瀏覽模擬的新聞提要，所有這些資料都是從 Web 服務擷取下來的。

若要瞭解背景運作方式，請使用網路監視器。

---

## A.13 MobileMediaAPI

MobileMediaAPI 應用程式包含四個展示工具組多媒體功能的 MIDlet。這一節描述這些 MIDlet，並包含從您的應用程式使用多媒體的其他資訊。

## A.13.1 Simple Tones

Simple Tones 範例示範如何使用互動合成鈴聲。選取一個範例，然後按一下右下方的 [ 播放 ]。

- Short Single Tone 和 Long Single Tone 使用 `Manager.playTone()` 播放不同音高的鈴聲。
- [ 短 MIDI ] 事件在互動式 MIDI 裝置 ( 定位器 "device://midi" ) 上播放一段和弦。MIDletControl 的 `shortMidiEvent()` 方法用來觸發和弦的音符。
- 若要執行 MMAPI Drummer 示範，請按一下或鍵入數字鍵 (0-9)。各數字鍵所播放的聲音不同。

## A.13.2 Simple Player

Simple Player 應用程式會示範模擬器的音效與視訊功能範圍。它包含各種不同格式的範例檔，而且可以從模擬器的永久性儲存裝置或從 HTTP URL 播放檔案。

播放器部分使用通用的 `javax.microedition.media.Player` 介面。播放器會顯示長度、媒體時間和執行媒體檔的控制項。如果檔案提供中介資料，此播放器可讓您檢視這些資訊，例如作者與標題。對於 MIDI 檔案，如果檔案提供卡拉 OK 文字，則播放時螢幕上會顯示文字。如果提供圖形使用者介面控制項，也可在顯示螢幕上進行檢視。若要存取這些控制項，請在 [Simple Player] 中選取其中一個媒體範例，然後按下 [ 功能表 ] 按鈕，以檢視並選取需要的指令。

選取 [Simple Player]，然後按一下 [ 啟動 ]。示範包含下列媒體範例：

- Bong 會播放一個短 WAV 檔案。您可以依照本文件稍後描述的方式，調整某些播放功能。顯示上會顯示聲音的長度，格式為「分:秒:十分之一秒」，例如 00:17:5。此音效範例是 MIDlet 套裝軟體 JAR 檔案中的一個資源檔案。
- MIDI Scale 會播放音階範例。顯示上會顯示所選音樂檔的標題、歌曲長度、已播放時間及目前的音樂速度 ( 以每分鐘節拍數 (bpm) 為單位 )。此 MIDI 檔存放於 MIDlet 套裝軟體 JAR 檔案中。
- Simple Ring Tone 會播放一小段貝多芬第五號交響曲。顯示上會顯示所選音樂檔的標題、歌曲長度、已播放的秒和十分之一秒數、以及目前的音樂速度 ( 以每分鐘節拍數 (bpm) 為單位 )。此 ringtone 檔 (.jts 格式) 存放於 MIDlet 套裝軟體 JAR 檔案中。
- WAV Music 會播放簡短的音效檔。顯示上會顯示音效檔的標題、音效長度、已播放的時間以及播放速率百分比。此 WAV 檔從 HTTP 伺服器擷取而來。
- MIDI Scale 會播放從 HTTP 伺服器擷取來的 MIDI 檔。
- Animated GIF 範例會展示從 1 數到 5 的 GIF 動畫。檔案存放於 MIDlet 套裝軟體 JAR 檔案中。
- Audio Capture ( 從預設裝置 ) 可讓您從麥克風或連接的裝置擷取音效。聲音會擷取下來，並從喇叭播放出來。若要避免回音，請使用耳機。

- Video Capture Simulation 會模擬檢視輸入視訊，例如使用配備相機的裝置。
- MPEG1 Video [http]。可播放位於 <http://java.sun.com/products/java-media/mma/media/test-mpeg.mpg> 的 MPEG 視訊。
- [enter URL] 可讓您播放來自任意 HTTP 伺服器的媒體檔。在插入點處鍵入有效的 URL (例如，<http://java.sun.com/products/java-media/mma/media/test-wav.mpg>)，然後按一下 [確定] 以播放檔案。如果要開啓 HTTP 目錄，從中選取媒體，請務必確定在 URL 末端加上斜線。

此外，Simple Player 會剖析鈴聲文字傳輸語言 (RTTTL) 格式的鈴聲。如需 RTTTL 的詳細資訊，請參閱 <http://www.convertyourtone.com/rtttl.html>。

[Simple Player] 包含控制媒體播放的通用指令集。這些指令列在 [Simple Player] 功能表上，而且有些指令有相關的小鍵盤按鈕。下表描述這些指令。

**表格 A-3** Simple Player 指令

指令	按鍵	描述
靜音 / 取消靜音	0	關閉聲音，但重新繼續檔案。此指令切換成 [取消靜音]。
音量	* 與 #	提高或降低聲音大小。
中介資料		顯示媒體檔案所提供的資訊，例如版權資訊、標題與曲目清單等。
在 5 秒內停止		播放過程中，若設定此指令，要在 5 秒鐘內暫停播放音效。
速率	4 與 6	改變播放速率。
音樂速度		加快或放慢鈴聲序列或 MIDI 檔案的音樂速度。
音高	向上鍵 頭與向 下鍵頭	降低或升高 MIDI 檔的音調。
開始 / 停止錄音		錄製音效播放。在執行模擬器的目錄中建立一個檔案，內含錄製的音效。如果不指定檔案名稱，則建立的檔案名稱爲 recording.wav。此指令會切換成 [停止錄音]。
逐步執行螢幕格	7 與 9	在視訊檔中，一次往前或往後跳動一格。
播放 / 停止	[2] 與 [選取]	開始或停止媒體。
迴圈模式		在播放完成後，立即播放音效檔。執行一次 [迴圈模式] 就會播放一次音效檔。按第二次會將檔案播放三次。按第三次則會一再重複播放檔案。按第四次則恢復單次播放。
跳過	1 與 3	往前或往後跳過媒體檔長度的 5%。聲音會與視訊同步。
倒轉		回到音效播放起點。
停止並倒轉	5	停止播放，並倒轉回起點。
快速說明		顯示指令與小鍵盤按鈕清單。

依 [Simple Player] 所播放的媒體類型而定，有些指令可能不適用。此外，有些指令可以用小鍵盤按鈕呼叫。下表描述可用的指令、其小鍵盤對等按鍵和 MMAPi 中相關的類別。

請注意，Simple Player 應用程式本身提供一份簡短的指令清單，和對應的小鍵盤按鈕。只要從功能表選擇 [快速說明] 指令即可取得。

## A.13.3 Video

Video 應用程式演示模擬器如何播放 GIF 動畫檔與擷取視訊。在配備相機的實際裝置上，視訊擷取可用來告訴使用者，透過相機所能看到的內容。

使用 Form Item 或 Canvas 即可實作 GIF 動畫與視訊擷取。Video 示範包含所有可能性。Animated GIF - Form [jar] 會以 Form Item 方式顯示 GIF 動畫。表單也包含有關播放的一些資訊，包括目前時間。選擇 [快照] 指令可拍下執行中動畫的快照。快照將存放於跟在 GIF 動畫後面的表單中。

- Animated GIF - Canvas [jar] 會以 Canvas 格式顯示 GIF 動畫。簡單的快訊會顯示動畫的進度。選擇 [快照] 可取得目前外觀的靜態影像。顯示上會短暫顯示快照，然後回到動畫。
- Video Capture - Form 會模擬從相機或其他來源擷取視訊，並以 Form 中的 Item 方式顯示。選擇 [快照] 指令可拍下擷取視訊的快照。快照將存放於跟在視訊擷取後面的表單中。
- Video Capture - Canvas 會模擬從相機或其他來源擷取視訊，並以 Canvas 方式顯示。選擇 [快照] 可取得目前外觀的靜態影像。顯示幕上會短暫顯示快照，然後回到視訊擷取。
- MPEG1 Video - Form ; MPEG1 Video - Canvas

MPEG1 應用程式從 Web 獲得 MPEG，因此如果您在防火牆後方，則必須配置模擬器的代理伺服器設定。

選擇 [編輯] > [偏好]，然後選取 [網路配置]。核取 [使用代理伺服器]。填入代理伺服器位址欄位以及連接埠號。對於此示範，請選取 [HTTP/1.0]。

當您播放此示範時，必須等幾秒鐘讓 WTK 獲得資料。MPEG1 示範分別與 Video Capture - Form 和 Video Capture - Canvas 具有相同的運作方式。

## A.13.4 暫停音效測試

此 MIDlet 用於示範暫停的 MIDlet 未停止執行中的播放器時，Sun Java™ Wireless Toolkit for CLDC 要如何向您發出警告。啟動 MIDlet 之後，請選擇 [播放] 指令以開始播放某些音效。螢幕會顯示狀態「正常」或「不正常」。

從模擬器視窗的功能表中選擇 [MIDlet] > [暫停]。就如預期的，MIDlet 會暫停且工具組主控台上不會顯示訊息。從模擬器視窗的功能表中選擇 [MIDlet] > [重新繼續]。

接著選擇 [ 不正常 ] 指令。再次暫停 MIDlet。在工具組主控台中，您會看到以下警告：  
An active media (subtype Player) resource was detected while the MIDlet is paused. (MIDlet 處於暫停狀態時偵測到使用中的媒體 [ 子類型播放器 ] 資源。) Well-behaved MIDlets release their resources in pauseApp(). ( 正常 MIDlet 釋放在 pauseApp() 中的資源。)

## A.13.5 MobileMediaAPI 的屬性

MobileMediaAPI 應用程式有下列屬性，是您在專案設定對話方塊的 [ 用戶定義的 ] 標籤中可以修改的項目：

表格 A-4 MMAPI 特定 MIDlet 屬性的說明

屬性	描述
PlayerTitle- <i>n</i>	第 <i>n</i> 個要用 [Simple Player] MIDlet 播放的媒體標題名稱。
PlayerURL- <i>n</i>	第 <i>n</i> 個要用 Simple Player MIDlet 播放的媒體標題 PlayerTitle- <i>n</i> 之位置。
VideoTest- <i>n</i>	第 <i>n</i> 個要用 Video 應用程式播放的媒體標題名稱。
VideoTest-URL <i>n</i>	第 <i>n</i> 個要用 Video 應用程式播放的媒體標題 VideoTest- <i>n</i> 之位置。

## A.14 網路示範

此示範有兩個 MIDlet：Socket Demo 和 Datagram Demo。每個示範都需要執行兩個模擬器實例，以便模擬伺服器 and 用戶端的關係。

### A.14.1 Socket Demo

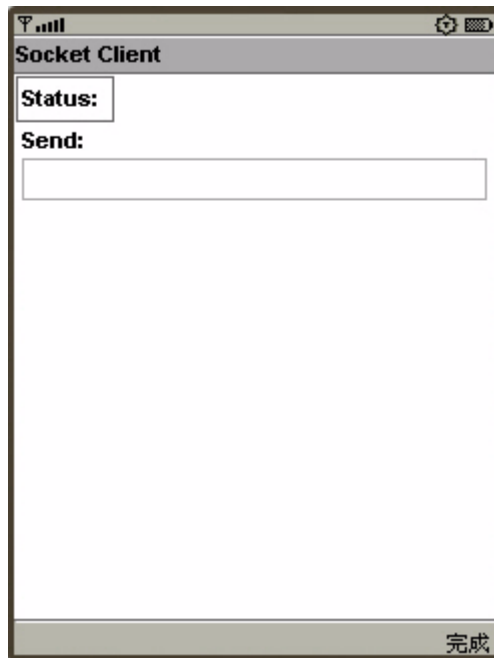
執行兩個模擬器實例。一個作為通訊端伺服器，另一個作為通訊端用戶端。

在第一個模擬器中，啟動應用程式，然後選取對等的伺服器。選擇 [ 啟動 ]。此模擬器說明此示範要透過網路傳送和接收資料，並詢問「Is it OK to use network? ( 是否可以使用網路? )」選擇「是」。通訊端伺服器會顯示螢幕，表示正在等待連線。

在第二個模擬器中啟動應用程式、選取對等的用戶端，然後選擇 [ 啟動 ]。此模擬器說明此示範要透過網路傳送和接收資料，並詢問「Is it OK to use network? ( 是否可以使用網路? )」選擇「是」。通訊端用戶端會顯示螢幕，表示已連接到伺服器。使用下導覽箭頭反白顯示 [ 傳送 ] 方塊。在 [ 傳送 ] 方塊中鍵入訊息，然後選擇 [ 傳送 ] 軟鍵。

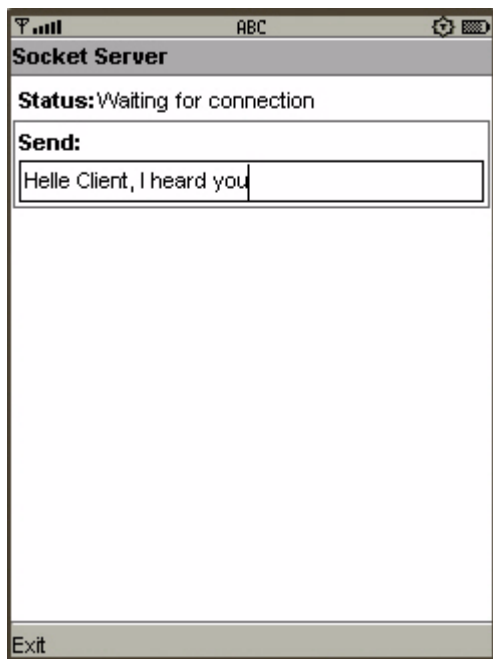
例如在用戶端的 [ 傳送 ] 方塊中鍵入 [Hello Server] ( 請參考圖 A-22 )。選擇 [ 傳送 ] 軟鍵。傳送期間模擬器會啟動藍燈。

圖 A-22 從通訊端用戶端傳送訊息



在執行通訊端伺服器的模擬器上，狀態為：Message received - Hello Server ( 訊息已收到 — 您好，伺服器 )。您可以使用向下箭頭移動到 [ 傳送 ] 方塊並鍵入回覆。例如，Hello Client, I heard you ( 您好，用戶端，我已收到您的訊息 )。選取 [ 傳送 ]。請參閱圖 A-23。

圖 A-23 伺服器顯示已收到的訊息和要傳送的訊息



回到通訊端用戶端，狀態顯示伺服器傳送的訊息。在傳送新訊息之前，[ 傳送 ] 方塊內都會包含您之前傳送的訊息。

## A.14.2 Datagram Demo

此示範類似 Socket Demo。

執行兩個模擬器實例。一個作為資料封包伺服器，另一個作為資料封包用戶端。

在第一個模擬器中，啟動 Datagram Demo，然後選取對等的伺服器。選擇 [ 啟動 ]。此模擬器說明此示範要透過網路傳送和接收資料，並詢問「Is it OK to use network? (是否可以使用網路?)」選擇「是」。最初，資料封包伺服器狀態為 [ 等待連線 ]，並且 [ 傳送 ] 為空。

在第二個模擬器中，啟動 Datagram Demo，選取對等的用戶端，然後選擇 [ 啟動 ]。此模擬器說明此示範要透過網路傳送和接收資料，並詢問「Is it OK to use network? (是否可以使用網路?)」選擇「是」。資料封包用戶端的狀態為：已連線到伺服器。使用下導覽箭頭反白顯示 [ 傳送 ] 方塊。在 [ 傳送 ] 方塊中鍵入訊息，然後選擇 [ 傳送 ] 軟鍵。例如，鍵入 [Hello datagram server (您好，資料封包伺服器)]。

在執行資料封包伺服器的模擬器上，狀態顯示為：Message received - Hello datagram server (訊息已收到 — 您好，資料封包伺服器)。您可以使用下箭頭移動到 [ 傳送 ] 方塊並鍵入對用戶端的回覆。

資料封包用戶端的狀態欄位顯示已從伺服器收到的訊息。[ 傳送 ] 方塊包含您傳送的最後一條訊息。



## A.15 ObexDemo

此應用程式會展示如何使用 OBEX API，在模擬器實例之間傳輸影像檔。此示範會展示透過模擬的紅外線連線使用 OBEX 的方式。

執行兩個模擬器實例。一個偵聽內送的連線，另一個則嘗試傳送影像。在第一個模擬器中，啟動應用程式後選擇 [Obex 示範]，然後選擇 [接收影像]。此模擬器說明 OBEX 連線允許其他裝置與您通話，並詢問「Is it OK to make the connection? (是否可以建立連線?)」選擇「是」。偵聽程式模擬器會顯示螢幕，指出它正在等待內送的連線。

在第二個模擬器中(寄件者)，啟動 [Obex 示範]，然後選擇 [傳送影像]。就會顯示影像清單。選取一個影像，並選擇 [傳送]。此模擬器說明示範要建立外送用戶端連線，並會詢問是否允許。選擇「是」。「傳送影像」公用程式會上傳影像。

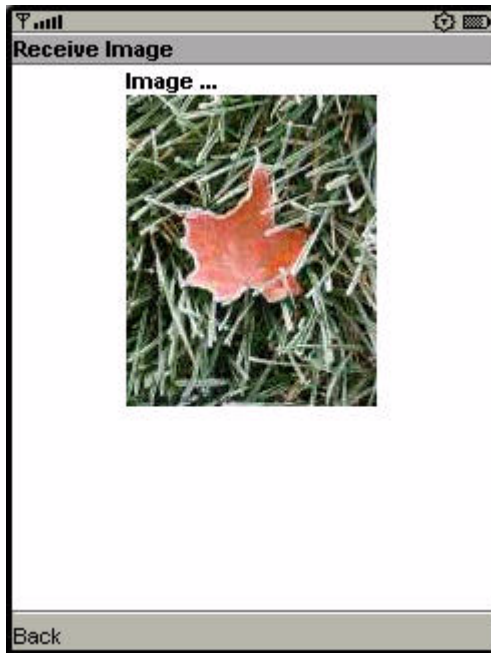
在偵聽模擬器中，公用程式會顯示有關內送影像的資訊，並詢問「Would you like to receive it? (您是否願意接收?)」請參閱圖 A-24。

圖 A-24 提示接受連線的偵聽程式



選擇「是」。選取的影像就會透過模擬的紅外線連結傳輸，顯示在第一個模擬器上。請參閱圖 A-25。

圖 A-25 成功傳輸影像



---

## A.16 PDAPDemo

PDAPDemo 展示如何使用 PIM 與 FileConnection API (這是 JSR 75 規格的一部分)。

### A.16.1 瀏覽檔案

如果您尚未完成此作業，若要執行檔案瀏覽器，將需要為 MIDlet 提供適當的安全授權。選擇 [編輯] > [偏好]。按一下 [安全性] 標籤。將 [安全網域] 變更為最大值，再按下 [確定]。

接著開啓並執行 PDAPDemo 專案。啓動 FileBrowser MIDlet。您會看到目錄清單，並瀏覽可用的目錄和檔案。預設狀況下有一個目錄 root1。

圖 A-26 瀏覽檔案



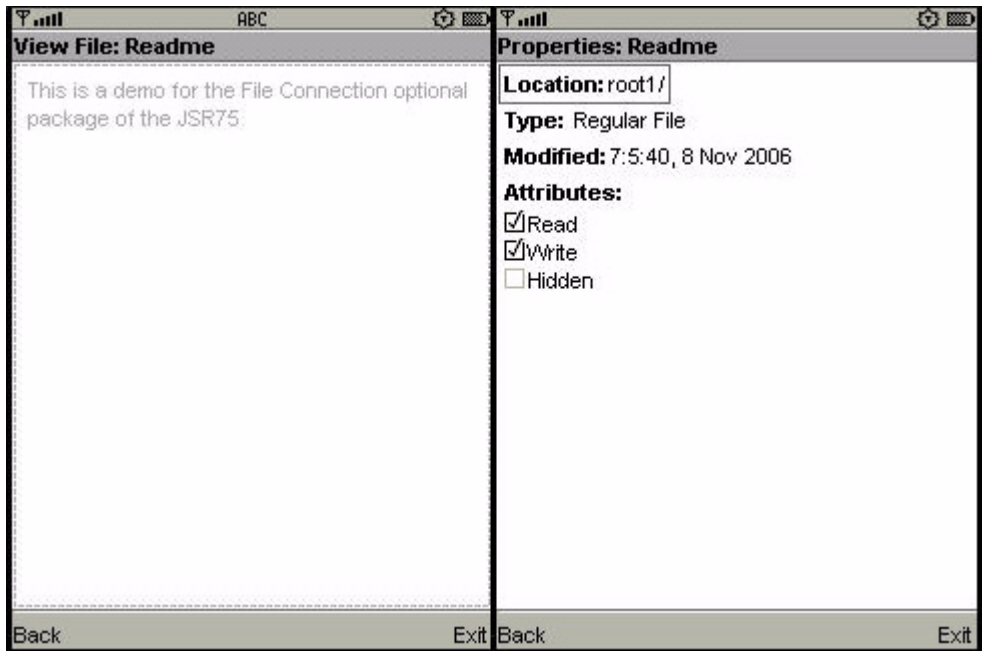
選取目錄，再按下 [ 選取 ] 按鈕以輸入該目錄。

圖 A-27 root1 目錄的內容



使用示範中的指令，您可以檢視檔案，或查看檔案特性。請嘗試選取檔案，並從功能表選擇 [ 特性 ] 或 [ 檢視 ]。

圖 A-28 檢視檔案內容和檔案特性



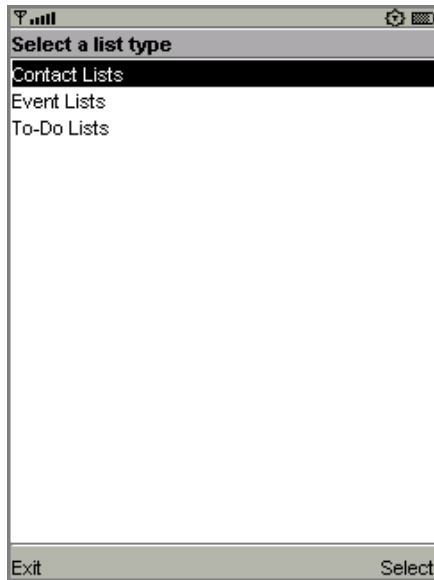
實際的檔案位於 `workdir\appdb\DefaultColorPhone\filesystem` ( 假設您使用的是 `DefaultColorPhone` 模擬器面板 )。您可以隨意增加檔案與根目錄，就可以在 JSR 75 File API 中看到這些項目。如需詳細資訊，請參閱[第 10 章](#)。

## A.16.2 PIM API

JSR75 PIM API 範例示範如何存取個人資訊，例如連絡人清單、行事曆與待辦事項清單。啟動範例之後，從主功能表選擇一種清單。

在這個應用程式範例中，每種清單的運作方式都一樣，而且每種清單類型都只包含一個清單。例如，如果選擇 [ 連絡人清單 ]，就只有一個 **Contacts** 連絡人清單。[ 事件清單 ] 只包含一個名為 **Events** 的清單，而 [ 待辦清單 ] 只包含一個名為 **To do** 的清單。

圖 A-29 選擇清單類型



選取清單類型，並選擇特定清單後，便可以檢視清單中的所有項目。如果這是您第一次執行本範例，清單可能是空的。

若要增加項目，請從功能表選擇【新增】。應用程式會提示您輸入項目的格式化的名稱。您可以用功能表中的【增加欄位】將更多其他資料欄位增加到此項目。就會顯示欄位名稱清單。挑選一個欄位，然後輸入新欄位的值。

圖 A-30 增加連絡人欄位



若要儲存清單項目，請從功能表選擇 [ 確定 ] ( 選項 3)。

您可以選擇 [ 上一步 ] 指令，回到清單。剛建立的項目就會顯示在清單中。

您建立的項目會以標準 vCard 或 vCalendar 格式儲存在 *workdir\appdb\skin\pim* 目錄中。如需詳細資訊，請參閱第 10 章。

PIM API 允許匯出標準格式的連絡人、行事曆與待辦事項。確實的格式須視清單類型而定。當您檢視任何清單中的項目時，功能表會包含用於檢視匯出項目的指令。

例如，當您檢視連絡人清單項目時，功能表會包含 [ 顯示 vCard]。當您選擇此指令時，匯出的項目就會顯示在螢幕上。行事曆項目與待辦事項都會匯出為 vCalendar。

---

## A.17 SATSADemos

SATSADemos 包含 SATSA (Security and Trust Services API) 的示範。大部分示範顯示如何與智慧卡通訊。模擬器可使用通訊端協定與模擬的智慧卡通訊。智慧卡模擬器 (cref) 包含在工具組中。如需詳細資訊，請參閱第 14 章。

以下各節中包含此示範的每個功能表選擇的說明。對於每個示範，在啟動模擬器之前一定要執行以下作業：

- 從指令行執行 cref 的實例。
- 一定要將安全網域設定為最大值。

## A.17.1 APDUMIDlet

此 MIDlet 示範使用應用程式協定資料單元 (APDU；小的資料封包) 與智慧卡進行通訊。APDUMIDlet 需要找到兩個模擬的智慧卡。您可以使用 cref (Java Card Development Kit 的一部分) 執行智慧卡模擬器。

Mohair 應用程式包含預建立的記憶體影像，您可以將這些影像與 cref 配合使用。這些記憶體影像包含與 Mohair 相互作用的 Java Card 應用程式。這些記憶體影像位於 Mohair 專案的根目錄中。

在 Windows 上，依照如下方式啟動 cref 的兩個實例，分別對應每個模擬卡片槽 (假設目前的目錄為工具組安裝目錄)：

```
start bin\cref -p 9025 -i apps\SATSA Demos\demo2.eeprom
start bin\cref -p 9026 -i apps\SATSA Demos\demo2.eeprom
```

在 Linux 上，您可以使用：

```
toolkit/bin/cref -p 9025 -i apps/SATSA Demos/demo2.eeprom
toolkit/bin/cref -p 9026 -i apps/SATSA Demos/demo2.eeprom
```

請注意，連接埠號 (此範例中的 9025 和 9026) 必須符合您在 SATSA 偏好中指定的連接埠號，如第 14 章所述。而且，確定使用 demo2.eeprom 的正確路徑。

執行兩個智慧卡模擬器後，您便可執行 APDUMIDlet。

## A.17.2 SATMIDlet

SATMIDlet 示範智慧卡通訊與 APDU 通訊間的細微差異。

若要設定模擬的智慧卡，請使用 cref，與您對 APDUMIDlet 執行的作業非常相像。此次您不必指定連接埠號，而且記憶體影像不同：

---

**Windows:** start bin\cref -i apps\SATSA Demos\sat.eeprom

**Linux:** toolkit/bin/cref -i apps/SATSA Demos/sat.eeprom

---

執行智慧卡模擬器時，您可以執行 SATMIDlet 以與智慧卡應用程式進行通訊。

## A.17.3 CryptoMIDlet

CryptoMIDlet 示範 SATSA 的一般加密功能。它不會以任何方式與智慧卡互動。

## A.17.4 MohairMIDlet

MohairMIDlet 具有兩項功能。第一，尋找插槽、顯示所有可用的卡片槽。每個插槽均有一個數字，後跟「C」或「H」，分別表示冷插拔或熱插拔插槽。檢視插槽之後，請選取 [ 上一步 ] 以返回第一個螢幕。

MohairMIDlet 的第二部分 (SATSA-PKI 簽署測試) 使用智慧卡產生數位簽名。與之前的示範相同，您需要使用正確的記憶體影像執行 cref，才能準備從 MohairMIDlet 連線。在安裝目錄中鍵入以下路徑：

---

**Windows:** start bin\cref -i apps\SATSA Demos\sat.eeprom

**Linux:** workdir/bin/cref -i apps/SATSA Demos/sat.eeprom

---

在模擬器中，高亮度顯示 [SATSA-PKI 簽署測試] 並選擇 [ 選取 ]。會顯示以下確認訊息：

```
[This certificate will be used: ( 將使用此憑證: ) MohairAuth
(MohairAuth)]
```

選取 [ 確定 ] 軟鍵。

對於 PIN 1，請鍵入：1234

選取 [ 確定 ] 軟鍵。會顯示以下確認訊息：

```
This string will be signed: ( 此字串將接受簽署: ) JSR 177 Approved
(JSR 177 已同意)
```

選取 [ 確定 ] 軟鍵。會顯示以下確認訊息：

```
[This certificate will be used: ( 將使用此憑證: ) MohairAuth
(MohairAuth)]
```

選取 [ 確定 ] 軟鍵。

對於不可拒絕之金鑰 1 PIN，請鍵入：2345

---

## A.18 SATSAJCRMIDemo

此應用程式包含單一 MIDlet (JCRMIMIDlet)，可展示如何使用 Java Card RMI ( 方便使用智慧卡的遠端物件協定 ) 與智慧卡應用程式進行通訊。與 SATSA Demos 中的某些 MIDlet 相同，您需要使用適當的記憶體影像啟動 cref：

---

**Windows:** start bin\cref -p 9025 -i apps\SATSA Demos\demo2.eeprom

**Linux:** workdir/bin/cref -i apps/SATSA Demos/demo2.eeprom

---

現在執行 JCRMIMIDlet 以瞭解您的應用程式如何才能與智慧卡上的分散式物件進行通訊。



---

## A.19 SIPDemo

此應用程式是使用 SIP (JSR 180) 直接在兩個裝置間進行通訊的簡單範例。通常裝置使用具有代理伺服器的 SIP，設定某種直接通訊。如需與代理伺服器相關的更多完整範例，請查考 GoSip。

若要瞭解 SIPDemo 的工作方式，請執行模擬器的兩個實例。在第一個模擬器中，選擇 [ 接收訊息 ]。您可以使用預設連接埠 (5070)，然後選擇 [ 接收 ]。第一個模擬器正在偵聽內送的訊息。

在第二個模擬器中，選擇 [ 傳送訊息 ]。填入收件者、連接埠號、主旨和訊息的值，或接受預設值，然後選擇 [ 傳送 ]。您的訊息將顯示在第一個模擬器中。第一個模擬器的回應會顯示在第二個模擬器中。

開啓網路監視器後再嘗試一次。您將在網路監視器的 [SIP] 標籤中看到模擬器之間的通訊。

---

## A.20 SVGContactList

此應用程式使用不同的面板顯示相同的連絡人清單資訊和新聞標題。這些面板具有不同的顏色和字型。

選取 [SVGContactlist(skin 1)] 或 [SVGContactlist(skin 2)]，然後按一下 [ 啓動 ]。

使用向上、向下箭頭導覽連絡人清單。反白顯示的名稱標記有特殊的字元 (> 或一個點)，並且會以較大的字型顯示。

圖 A-31 使用面板 2 顯示的連絡人清單



按下選取按鈕即可瞭解有關反白顯示的名稱的更多資訊。

圖 A-32 連絡人清單詳細資訊



再次按下 [ 選取 ] 以返回到連絡人清單。

---

## A.21 SVGDemo

此套裝軟體包含的 MIDlet，示範 JSR 226 Scalable 2D Vector Graphics API for J2ME 的不同使用方法。此 API 提供載入操作、描繪和播放 SVG 內容的方式。

Scalable Vector Graphics (SVG) 1.1 規格定義以 XML 描述二維圖形的語言。完整規格可在 <http://www.w3.org/TR/SVG11/> 中找到。

SVG Tiny (SVGT) 屬於 SVG 的一部份，適合類似行動電話的小型裝置。請參閱 <http://www.w3.org/TR/SVGMobile/>。SVG Tiny 是一種精簡但功能強大的 XML 格式，用於描述豐富的互動式 2D 動畫內容。圖形元素可進行邏輯分組並以 SVG 標記標識。

### A.21.1 SVG 瀏覽器

SVGBrowser MIDlet 顯示常駐電話檔案系統的 SVG 檔案。執行此示範之前，請將 SVG 檔案置於目錄 *workdir\appdb\DefaultColorPhone\filesystem\root1* 中。

啟動此示範。應用程式將顯示 *root1* 的內容。選取您的 SVG 檔並選擇 [ 開啓 ] 軟鍵。

### A.21.2 描繪 SVG 影像

[ 描繪 SVG 影像 ] 會從檔案中載入 SVG 影像加以描繪。查看示範程式碼就能看出影像大小是即時調整，以精確符合顯示區域。輸出是清晰且鮮明的。

### A.21.3 播放 SVG 動畫

此應用程式播放描繪萬聖節賀卡的 SVG 動畫。按下 8 播放，按下 5 開啓，按下 0 停止。如果您按下 8，按下 5 會重新繼續動畫。如果您按下 0，按下 5 會從頭開始播放動畫。

SVG 檔描述各個影像元素如何隨著時間變化，以形成此短動畫效果。

在以下程式碼範例中，JSR 226 `javax.microedition.m2g.SVGImage` 類別用於載入 SVG 資源。然後，`javax.microedition.m2g.SVGAnimator` 類別可處理複雜的 SVG 動畫，並提供播放動畫的 `java.awt.Component` 或 `javax.swing.JComponent`。SVGAnimator 類別提供播放、暫停和停止動畫的方法。

```
import javax.microedition.m2g.ScalableGraphics;
import javax.microedition.m2g.SVGImage;

...
String svgURI = ...;
SVGImage svgImage = (SVGImage) SVGImage.createImage(svgURI, null);
SVGAnimator svgAnimator = SVGAnimator.createAnimator(svgImage);

// If running a JSE applet, the target component is a JComponent.
JComponent svgAnimationComponent = (JComponent)
svgAnimator.getTargetComponent();
...

svgAnimator.play();
...
svgAnimator.pause();
...
svgAnimator.stop();
```

## A.21.4 從頭開始建立 SVG 影像

此示範使用 API 呼叫生成影像。它可建立一個空 `SVGImage` 然後加入圖形內容，最後顯示該內容。

## A.21.5 跳跳球

跳跳球播放一個 SVG 動畫。按下 8 播放，按下 5 開啓，按下 0 停止。如果您按下 8，按下 5 會重新繼續動畫。如果您按下 0，按下 5 會從頭開始播放動畫。

## A.21.6 最佳化的功能表

在此示範中，選取的圖示有黃色的邊界。當您移動到新圖示時，該圖示會變為選取的狀態，之前的圖示會變為未選取的狀態。如果您導覽至圖示網格之外，則會進行循環選取。也就是說選取某列的最後一個圖示時，向右移動的話就會選取同一列的第一個圖示。

此示範說明了合併 UI 標記和 Java 可帶來的靈活性：豐富的功能集（圖形、動畫和高階 2D 描繪），在圖形操控、預描繪或播放方面也極具靈活性。

在此範例中，圖形藝術家提供的 SVG 動畫，定義了功能表圖示的轉換狀態（從未選取的狀態轉換為選取的狀態）。此程式使用 JSR 226 API 將每個圖示的動畫序列分別描繪至螢幕外的緩衝區（以便之後加速描繪作業）。

緩衝處理讓 MIDlet 能適應裝置的顯示解析度（因為圖形是以 SVG 格式定義的），並且保持點陣圖的描繪速度。此外，MIDlet 仍會運用 SVG 的動畫功能。

定義功能表項目外觀及其動畫效果的任務（圖形藝術家和設計者的工作），和顯示功能表以及根據功能表選項啟動動作的任務（開發人員的工作）劃分極為清楚。只要藝術家和開發人員均遵守 SVG 文件結構慣例，這兩項任務便可獨立區分。

## A.21.7 照片裝飾程式

在此示範中，您可使用電話鍵為照片增加裝飾。這些鍵代表的值為：

- 
- 1 縮小鍵
  - 2 下一張照片鍵
  - 3 放大鍵
  - 4 說明鍵
  - 5 水平翻轉鍵
  - 6 垂直翻轉鍵
  - 7 逆時鐘旋轉鍵
  - 8 上一張照片鍵
  - 9 順時鐘旋轉鍵
  - # 顯示挑選器選項
- 

此示範提供 16 張照片供您裝飾。

使用 2 和 6 鍵向前或向後導覽照片。

若要裝飾，請按下 [#] 顯示挑選器。使用箭頭鍵反白顯示圖形物件。放大反白顯示的物件。按下 [選取] 以選擇目前的圖形，或按下箭頭鍵反白顯示不同的圖形。再次按下 [選取] 以將圖形增加到照片。當您增加裝飾時，會看到圖形上有一個紅色的 +。這表示已經加以選取，並且可以移動、調整大小和處理。

圖 A-33 含有選取引文的裝飾照片



使用導覽箭頭移動圖形。使用 1 縮小圖形，使用 3 放大圖形。使用 5 或 6 翻轉，使用 7 或 9 旋轉。如果對位置滿意，就按下 [ 選取 ]。請注意，會顯示綠色三角形。這是游標。使用導覽鍵在圖形周圍移動綠色三角形。游標位於物件上的時候，會以紅色方框高亮度顯示物件。按下 [ 選取 ]。紅色 + 代表已經選取物件。

圖 A-34 高亮度顯示的小鬍子



若要移除裝飾（一種特性），請選取一個物件，然後按一下 [ 功能表 ] 軟鍵。按下 [2] 以移除特性。

## A.21.8 定位服務

啓動應用程式。會顯示軟體資訊畫面（也用作說明）。最初的視圖爲舊金山漫步的行程地圖。海灣（以藍色表示）在螢幕的右側。按下 [1] 開始行程。應用程式放大您在地圖上的位置。帶路指引顯示在水平軸的白色方塊中。執行行程時，按下 [7] 可逆時鐘旋轉地圖。請注意，地圖旋轉後文字會顯示在垂直軸上。再次按下 7 以還原預設方位。按下 4 以顯示說明螢幕。

圖 A-35



---

## A.22 WMADemo

此應用程式展示如何傳送與接收 SMS、CBS 與 MMS 訊息。Sun Java™ Wireless Toolkit for CLDC 提供有彈性的模擬環境，以支援訊息傳送。訊息可以在模擬器實例之間交換，而且可以利用 WMA 主控台公用程式產生或接收。

因爲此範例利用推送註冊表，所以只用 [執行] 按鈕看不到所有功能。利用在實際裝置上安裝應用程式的方式，使用 [經由 OTA 執行] 功能將應用程式安裝到模擬器。如果不知道如何執行，請查閱「[第 2 章](#)」。

若要運用推送註冊表，請使用 WMA 主控台，傳送一則訊息給模擬器。若要啟動主控台，請選擇 [ 檔案 ] > [ 公用程式 ]。在 WMA 方塊中按一下 [ 開啓主控台 ] 按鈕，以啟動 WMA 主控台。

按一下 WMA 主控台視窗中的 [ 傳送 SMS...] 按鈕。選擇模擬器的號碼 ( 可能是 +5550000 )。如果不確定模擬器使用的號碼是什麼，請查看標題列。在 SMS 訊息視窗中選擇號碼，然後填入連接埠號 50000。再在 [ 訊息 ] 欄位鍵入文字訊息，並按一下 [ 傳送 ]。

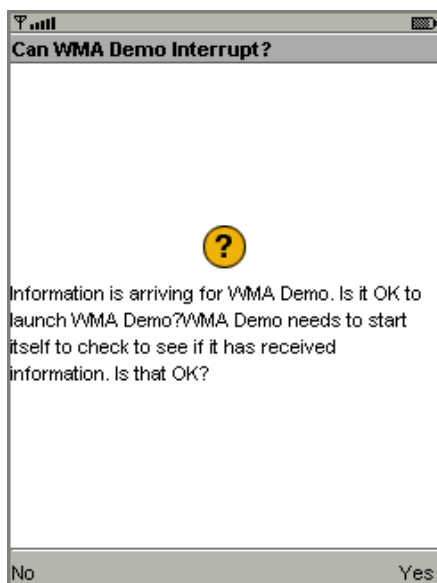
圖 A-36 傳送文字訊息



模擬器會詢問能否啟動 WMA Demo 應用程式。

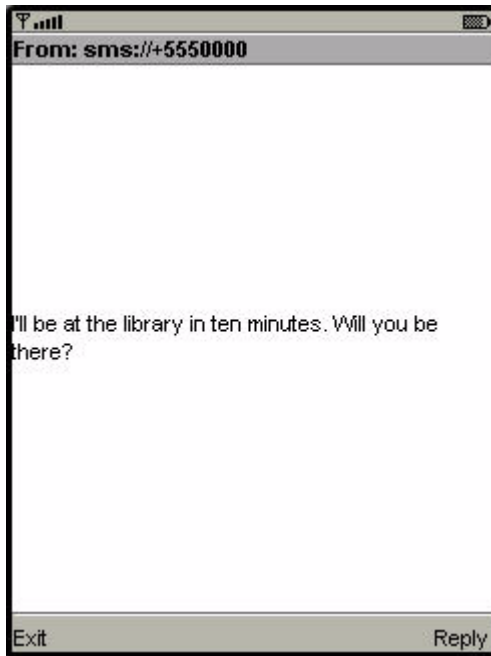


圖 A-37 推送註冊表訊息



選擇「是」。就會啟動 SMSReceive MIDlet，並立即顯示內送的 SMS 訊息。

圖 A-38 內送文字訊息



您也可以用 WMA 主控台傳送與接收 CBS 與 MMS 訊息。如需詳細資訊，請參閱 [第 7 章](#)。

---

**備註 –** 如果嘗試使用 WMA 主控台傳送文字訊息給 WMA Demo，務必將連接埠號指定為 50000。CBS 訊息則使用連接埠 50001。對於 MMS 訊息，請使用 `example.mms.MMSDemo` 做為應用程式 ID。

---

例如，若要從 WMA 主控台傳送 MMS 訊息給模擬器，必須確定已經依上述方式，使用 [經由 OTA 執行] 安裝 WMA Demo。啟動示範並選擇 [MMS 接收]。

在 WMA 主控台中，按一下 [傳送 MMS...]，以開啟 MMS 撰寫視窗。填入訊息主旨、應用程式 ID `example.mms.MMSDemo` 與執行中模擬器的電話號碼。

圖 A-39 指定 MMS 訊息的傳送位址

**+5550000 - 傳送訊息 - MMS**

標頭 部分

主旨: Millet's Garden by John Singer Sargent

應用程式 ID: example.ms.MMSDemo

收件者: mms://5550001

添加 (A) 移除 (R)

傳送 (S) 取消 (C)

接著，按一下 [ 部分 ] 標籤。WMA 主控台可讓您從硬碟中選擇要與 MMS 訊息一起傳送的檔案。按一下 [ 增加 ] 將檔案加入訊息中。使用檔案瀏覽器尋找要傳送的檔案，再按一下 [ 確定 ]。

圖 A-40 在 MMS 訊息中新增部分

**+5550000 - 傳送訊息 - MMS**

標頭 部分

檔案	內容位置	內容 ID	Mime 類型	編碼	大小
C:\millet-...	millet-gar...	0	image/jpeg	MS950	28521

總大小: 28521

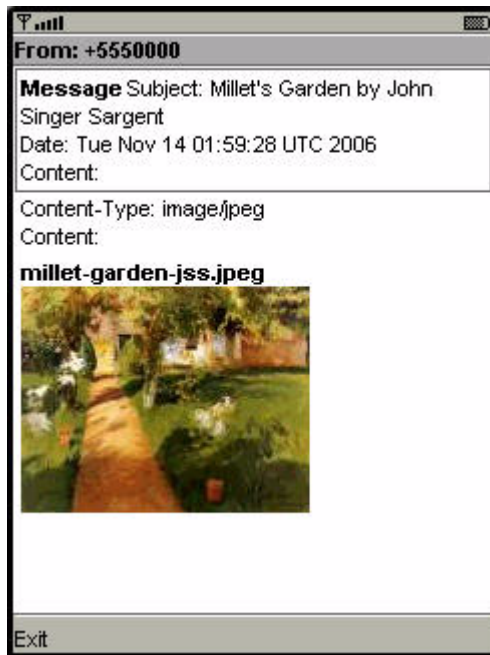
添加 (A) 移除 (R)

傳送 (S) 取消 (C)

按一下 [ 傳送 ] 以傳送訊息。

模擬器會詢問能否啟動 WMADemo。按一下 [ 是 ]。就會顯示影像及其資訊。

■ A-41 WMA Demo 接收影像



---

## 指令行參考

---

此附錄描述如何從指令行操作 Sun Java™ Wireless Toolkit for CLDC，並詳述生成和執行應用程式所需的步驟。此外，還描述 Sun Java™ Wireless Toolkit for CLDC 的憑證管理員公用程式（名稱爲 MEKeyTool），以及 MIDlet 簽署公用程式（其名稱爲 JadTool）（Java Application Descriptor Tool）。

---

### B.1 先決條件

從指令行生成和執行應用程式之前，請確認您的 Java SE 軟體開發工具組版本是 1.4.2 或更新版本。請確定您的路徑中包含 `jar` 指令。若要尋找開發工具組的版本，請於指令行執行 `jar` 指令，然後執行 `java -version`。

如需更多範例，請參閱示範應用程式 `bin` 目錄下的 `build.bat` 與 `run.bat` 兩個檔案。這些檔案位於：

---

<b>Windows:</b>	<code>toolkit\apps\demo\bin</code>
<b>Linux:</b>	<code>toolkit/apps/demo/bin</code>

---

`toolkit` 是 Sun Java™ Wireless Toolkit for CLDC 的安裝目錄，`demo` 是一個示範應用程式的名稱。

---

### B.2 開發周期

如需開發 MIDP 應用程式的完整說明，請參閱第 2 章。這一節描述如何從指令行完成開發周期的每一個步驟。

## B.2.1 生成

在使用者介面上，生成專案只需一個步驟。但在背景，其實需執行兩個步驟。首先，Java 原始碼檔案編譯成 Java 類別檔案。接著，預先驗證類別檔案，這表示準備好類別檔案，以用於 CLDC KVM。

使用 Java SE 開發工具組的 `javac` 編譯器編譯 Java 原始碼檔案。您可以使用現有的 Sun Java™ Wireless Toolkit for CLDC 專案目錄結構。您必須使用 `-bootclasspath` 選項，告訴編譯器使用 MIDP API，而且您可以使用 `-d` 選項，指示編譯器放置已編譯類別檔案的位置。

下列範例會示範如何編譯 MIDP 2.0 應用程式，把 `src` 目錄中的原始碼檔案，編譯成類別檔案放在 `tmpclasses` 目錄中。為清楚起見，每行多加了換行符號。

### Windows

```
javac
  -bootclasspath ..\..\lib\cldcapi10.jar;..\..\lib\midpapi20.jar
  -d tmpclasses
  src/*.java
```

### Linux

```
javac
  -bootclasspath ../../lib/cldcapi10.jar;../../lib/midpapi20.jar
  -d tmpclasses
  src/*.java
```

如果要使用工具組支援的選擇性 API，請把它們的 JAR 檔案加到 `-bootclasspath` 選項。

如需 `javac` 的詳細資訊，請參閱 Java SE 文件。

下一個步驟是預驗證類別檔案。在 Sun Java™ Wireless Toolkit for CLDC 的 `bin` 目錄中有一個方便的公用程式，叫做 `preverify`。`preverify` 指令的語法如下：

```
preverify [options] files | directories
```

部分選項如下：

`-classpath classpath`

指定要載入其類別的來源目錄或 JAR 檔案 ( 以分號區隔清單 )。

`-d output-directory`

指定輸出類別的目標目錄。預驗證之前，此目錄必須存在。如果未使用此選項，預先驗證器會將類別放置於 `output` 目錄中。

在編譯範例之後，使用下列指令驗證已編譯的類別檔案。同樣為清楚起見，每行多加了換行符號。

## Windows

```
preverify
  -classpath ../../lib/cldcapi10.jar;../../lib/midpapi20.jar
  -d classes
  tmpclasses
```

## Linux

```
preverify
  -classpath ../../lib/cldcapi10.jar;../../lib/midpapi20.jar
  -d classes
  tmpclasses
```

執行此指令後，完成預先驗證的類別檔案會放置於 `classes` 目錄中。如果應用程式使用 WMA、MMAPI 或其他版本的 CLDC 或 MIDP，請確定在 `classpath` 中包含相關的 `.jar` 檔案。

## B.2.2 封裝

若要封裝 MIDlet 套裝軟體，您必須建立一個清單檔案，再生成一個應用程式 JAR 檔案，最後建立一個 MIDlet 套裝軟體描述元。

依 MIDP 規格所指定，建立包含適當屬性的清單檔案。您可以使用任何文字編輯器，建立清單檔案。清單可能含有下列內容：

```
MIDlet-1: My MIDlet, MyMIDlet.png, MyMIDlet
MIDlet-Name: MyMIDlet
MIDlet-Vendor: My Organization
MIDlet-Version: 1.0
MicroEdition-Configuration: CLDC -1.0
MicroEdition-Profile: MIDP -2.0
```

建立 JAR 檔案，內含清單及套裝軟體的類別與資源檔案。若要建立 JAR 檔案，請使用 Java SE 軟體開發工具組隨附的 `jar` 工具。語法如下：

```
jar cfm file manifest -C class-directory . -C resource-directory .
```

引數如下：

- *file* — 要建立的 JAR 檔案。
- *manifest* — MIDlet 的清單檔案。
- *class-directory* — 包含應用程式類別的目錄。
- *resource -directory* — 包含應用程式資源的目錄。

例如，若要建立一個名為 `MyApp.jar` 的 JAR 檔案，其類別位於 `classes` 目錄中，而資源位於 `res` 目錄中，請使用下列指令：

```
jar cfm MyApp.jar MANIFEST.MF -C classes . -C res .
```

依 MIDP 規格所指定，建立包含適當屬性的 JAD 檔案。您可以使用任何文字編輯器來建立 JAD 檔案。此檔案必須使用 .jad 副檔名。

---

**備註** – 您必須將 MIDlet-Jar-Size 項目設定為上一個步驟所建立的 JAR 檔案大小。

---

例如，JAD 檔案可能包含下列內容：

```
MIDlet-Name: MyMIDlet
MIDlet-Vendor: My Organization
MIDlet-Version: 1.0
MIDlet-Jar-URL: MyApp.jar
MIDlet-Jar-Size: 24601
```

## B.2.3 執行

您可以從指令行執行模擬器。Sun Java™ Wireless Toolkit for CLDC 的 bin 目錄包含 emulator 指令。emulator 指令的語法如下：

`emulator options`

一般選項包括：

- `-help` – 顯示有效選項清單。
- `-version` – 顯示關於模擬器版本的資訊。
- `-Xquery` – 在標準輸出位元流上顯示模擬器面板資訊，然後立即結束。此資訊包括面板名稱、螢幕大小和其他功能。

與執行 MIDlet 套裝軟體有關的選項包括：

- `-Xdevice:skin-name` – 在使用指定面板名稱的模擬器上執行應用程式。如需面板名稱清單，請參閱第 4-1 頁的第 4.1 節「模擬器面板」。
- `-Xdescriptor:jad-file` – 在本機上使用指定的 JAD 檔案執行應用程式。
- `-classpath classpath` – 指定執行應用程式必需的程式庫的類別路徑。在本機上執行應用程式時，應使用此選項。
- `-Dcom.sun.midp.io.http.proxy` – 於執行期間設定 HTTP 和 HTTPS 代理伺服器。例如：  
`-Dcom.sun.midp.io.http.proxy=proxy-host:proxy-port`
- `-Dcom.sun.midp.midlet.platformRequestCommand` – 指定應用程式呼叫 URL 時要使用的瀏覽器。例如：  
`-Dcom.sun.midp.midlet.platformRequestCommand=browser`

範例呼叫可能類似於：

```
emulator -Dcom.sun.midp.midlet.platformRequestCommand=firefox
-Xjam:install=URL-to-app-using-platformRequest-method
```



如果您希望每次都使用同樣的瀏覽器，可以將以下指令行加入至 `toolkit/lib/system.config`：

`com.sun.midp.midlet.platformRequestCommand`: 瀏覽器

**Windows**：在 Windows 中，如果未指定此參數，則將使用預設瀏覽器。

**Linux**：對於 Linux 則必須指定此參數，因為 Linux 系統一般沒有預設瀏覽器。如果未指定此參數，則當應用程式嘗試開啓 URL 時無任何反應。

- `-Xjam:command=application` — 使用應用程式管理軟體 (Application Management Software, AMS) 從遠端執行應用程式，以使用 OTA 供裝執行。如果未使用引數指定任何應用程式，將執行圖形 AMS。指令如下：

`install=jad-file-url | force | list | storageNames|`

使用指定的 JAD 檔案將應用程式安裝到裝置上。

以及

`run=[storage-name | storage-number]`

執行之前安裝的應用程式。應用程式是使用其有效的儲存裝置名稱或儲存裝置號碼來指定的。

`remove=[storage-name | storage-number | all]`

移除之前安裝的應用程式。應用程式是使用其有效的儲存裝置名稱或儲存裝置號碼來指定的。指定 `all`，則移除之前安裝的所有應用程式。

- `transient=jad-file-url` — 使用指定的 JAD 檔案來安裝、執行和移除應用程式。指定 `transient` 將使應用程式安裝並執行，然後再移除三次。

## B.2.4 除錯

您可以在模擬器使用下列選項，進行除錯與追蹤。

- `-Xverbose:trace-options` — 依照由逗號分隔的選項，顯示追蹤輸出，如下所示：
  - `gc` — 追蹤資源回收
  - `class` — 追蹤類別載入
  - `all` — 使用全部追蹤選項
- `-Xdebug` — 啓用執行階段除錯。也必須使用 `-Xrunjdw` 選項。
- `-Xrunjdw:debug-settings` — 依照由逗號分隔的除錯設定，啓動 Java 除錯線路協定階段作業。也必須使用 `-Xdebug` 選項。有效的除錯設定包括：
  - `transport=transport-mechanism` — 與除錯程式通訊時使用的傳輸機制。唯一支援的傳輸機制是 `dt_socket`。
  - `address=host:port` — 除錯程式連線的傳輸位址。您可以不提供 `host`。如果省略 `host`，即假設主機是 `localhost`。
  - `server={y|n}` — 啓動除錯代理程式做為伺服器。除錯程式必須連接到指定的連接埠。可能值為 `y` 與 `n`。目前，只支援 `y` (模擬器必須做為伺服器)。

## B.3 啓動工具組 GUI 元件

Sun Java™ Wireless Toolkit for CLDC 的元件全部都可以從指令行啓動。每個元件都位於工具組的 `bin` 目錄中。

表格 B-1 工具組元件指令

指令	描述
DefaultDevice	顯示對話方塊，讓您選擇預設模擬器面板
ktoolbar	啓動使用者介面。
prefs	啓動工具組偏好
utils	啓動工具組公用程式視窗

## B.4 設定模擬器偏好

您可以使用 `emulator` 指令的 `--Xprefs` 選項，從指令行變更模擬器偏好。格式如下：

`-Xprefs:filename`

提供的 *filename* 為特性檔案的完整路徑名稱，這個值會置換偏好對話方塊中的值。特性檔案可包含下表所述特性：

表格 B-2 模擬器偏好特性清單

特性名稱	特性說明與法定值
http.version	網路配置 > HTTP 版本 值：HTTP/1.1   HTTP/1.0
http.proxyHost	網路配置 > HTTP 位址 值：主機名稱
http.proxyPort	網路配置 > HTTP 連接埠 值：integer
https.proxyHost	網路配置 > HTTPS 位址 值：主機名稱
https.proxyPort	網路配置 > HTTPS 連接埠 值：integer

**表格 B-2** 模擬器偏好特性清單 (繼續)

特性名稱	特性說明與法定值
<code>kvem.memory.monitor.enable</code>	監視器 > 啟用記憶體監視器 值：true   false
<code>kvem.netmon.comm.enable</code>	監視器 > 啟用通訊監視 值：true   false
<code>kvem.netmon.datagram.enable</code>	監視器 > 啟用資料封包監視 值：true   false
<code>kvem.netmon.http.enable</code>	監視器 > 啟用 HTTP 監視 值：true   false
<code>kvem.netmon.https.enable</code>	監視器 > 啟用 HTTPS 監視 值：true   false
<code>kvem.netmon.socket.enable</code>	監視器 > 啟用通訊端監視 值：true   false
<code>kvem.netmon.ssl.enable</code>	監視器 > 啟用 SSL 監視 值：true   false
<code>kvem.profiler.enable</code>	監視器 > 啟用效能評測 值：true   false
<code>netspeed.bitpersecond</code>	效能 > 位元 / 秒組合方塊 值：integer
<code>netspeed.enableSpeedEmulation</code>	效能 > 啟用流量模擬 值：true   false
<code>screen.graphicsLatency</code>	效能 > 圖形基元等待時間 值：integer
<code>screen.refresh.mode</code>	效能 > 顯示更新 (單選按鈕) 值：default   immediate   periodic
<code>screen.refresh.rate</code>	效能 > 顯示更新 (滑動軸) 值：integer
<code>vmspeed.bytecodespermilli</code>	效能 > 啟用虛擬機器速度模擬 (核取方塊) 值：integer
<code>vmspeed.enableEmulation</code>	效能 > 啟用虛擬機器速度模擬 (滑動軸) 值：true   false
<code>storage.root</code>	儲存裝置 > 儲存裝置根目錄 值：String ( <i>appdb</i> 相對路徑)
<code>storage.size</code>	儲存裝置 > 儲存裝置大小 值：integer

**表格 B-2** 模擬器偏好特性清單（繼續）

特性名稱	特性說明與法定值
<code>mm.control.capture</code>	MMedia > 音效擷取 值：true   false
<code>mm.control.midi</code>	MMedia > MIDI 音調 值：true   false
<code>mm.control.mixing</code>	MMedia > 音效混合 值：true   false
<code>mm.control.record</code>	MMedia > 音效記錄 值：true   false
<code>mm.control.volume</code>	值：true   false
<code>mm.format.midi</code>	MMedia > MIDI 格式 值：true   false
<code>mm.format.video</code>	MMedia > 視訊格式 值：true   false
<code>mm.format.wav</code>	MMedia > WAV 音效格式 值：true   false
<code>wma.client.phoneNumber</code>	WMA > 下一個模擬器的電話號碼 值：integer
<code>wma.server.firstAssignedPhoneNumber</code>	WMA > 優先指定的電話號碼 值：integer
<code>wma.server.percentFragmentLoss</code>	WMA > % 隨機訊息分割遺失 值：integer
<code>wma.server.deliveryDelayMS</code>	WMA > 訊息分割傳送延遲 (ms) 值：integer

## B.5 使用安全功能

從指令行也能使用 Sun Java™ Wireless Toolkit for CLDC 完整的安全功能。您可以調整模擬器的預設保護網域、簽署 MIDlet 套裝軟體和管理憑證。

## B.5.1 變更模擬器的預設保護網域

若要調整模擬器的預設保護網域，請在 `emulator` 指令中使用下列選項：

```
-Xdomain domain-type
```

指定 MIDlet 套裝軟體的安全網域。網域類型包括 `untrusted`、`trusted`、`minimum` 和 `maximum`。

## B.5.2 簽署 MIDlet 套裝軟體

JadTool 是一個指令行介面，可依據 MIDP 2.0 規格，使用公開金鑰加密來簽署 MIDlet 套裝軟體。簽署 MIDlet 套裝軟體的過程，是將簽署人憑證與 JAR 檔案的數位簽名加入 JAD 檔案。JadTool 也可以簽署支付款項更新 (JPP) 檔案。

JadTool 只使用來自 Java SE 平台金鑰庫的憑證與金鑰。Java SE 軟體提供指令行工具 `keytool` 來管理 Java SE 平台金鑰庫。

JadTool 封裝在 JAR 檔案中。若要執行它，請開啓指令提示符號，將目前目錄切換至 `toolkit\bin`，然後輸入下列指令：

```
java -jar JadTool.jar command
```

指令如下：

- `-help`

顯示 JADTool 的使用方法說明。

- `-addcert -alias alias [-keystore keystore] [-storepass password] [-certnum number] [-chainnum number] [-encoding encoding] -inputjad | inputjpp input-file -outputjad | outputjpp output-file`

將指定金鑰庫中金鑰組的憑證加入 JAD 檔案或 JPP 檔案中。

- `-addjarsig -jarfile jarfile -keystore keystore -alias alias -storepass password -keypass password -inputjad input-jadfile -outputjad output-jadfile`

將指定 JAR 檔案的數位簽名加入指定的 JAD 檔案中。`-jarfile` 的預設值是 JAD 檔案中的 MIDlet-Jar-URL 特性。

- `-showcert (([-certnum number] [-chainnum number]) | -all [-encoding encoding] -inputjad filename)`

顯示指定 JAD 檔案中的憑證清單。

- `-addjppsig -alias alias -keypass password [-keystore keystore] [-storepass password] [-encoding encoding] -inputjpp filename -outputjpp filename`

將輸入 JPP 檔案的數位簽名加入指定的輸出 JPP 檔案中。

預設值如下所示：

- -encoding — UTF-8
- -jarfile - JAD 檔案中的 MIDlet-Jar-URL 特性
- -keystore — %HOMEPATH%\keystore
- -certnum — 1
- -chainnum — 1

## B.5.3 管理憑證

MEKeyTool 用於管理憑證授權單位 (CA) 的公開金鑰，因此在功能上類似 Java SE SDK 隨附的 keytool 公用程式。使用金鑰有助於透過 SSL 進行安全的 HTTP 通訊 (HTTPS)。

使用 MEKeyTool 之前，您必須先具備 Java Cryptography Extension 金鑰庫的存取權。您可以使用 Java SE keytool 公用程式建立一個。

### Windows

<http://java.sun.com/javase/6/docs/technotes/tools/windows/keytool.html>

### Linux

<http://java.sun.com/javase/6/docs/technotes/tools/solaris/keytool.html>

若要執行 MEKeyTool，請開啓指令提示符號，將目前目錄變更為 *toolkit\bin*，然後輸入下列指令：

---

**Windows :**    *toolkit\bin\mekeytool.exe -command*

**Linux :**        *toolkit/bin/mekeytool -command*

---

隨後為指令關鍵字。請注意，當 MEKeyTool 從安裝目錄執行時，預設鍵以及您所建立的鍵將在您的個人目錄 *workdir\appdb* 中常駐。

- -help

顯示 MEKeyTool 的使用方法說明。

- -import -alias *alias* [-keystore *JCEkeystore*] [-storepass *storepass*] -domain *domain-name*

使用指定的 Java Cryptography Extension 金鑰庫密碼，將公開金鑰從指定的 JCE 金鑰庫匯入 ME 金鑰庫。預設 ME 金鑰庫是 *workdir\appdb\\_main.mks*，而預設 Java Cryptography Extension 金鑰庫是 *user.home\keystore*。

■ -list

列出 ME 金鑰庫中的金鑰，包括各金鑰的所有者與有效期限。ME 金鑰庫是 `workdir\appdb\_main.mks`。

■ -delete (-owner *owner* | -number *key-number*)

從指定所有者的指定 ME 金鑰庫中刪除一個金鑰。ME 金鑰庫是 `workdir\appdb\_main.mks`。

---

**備註** – Sun Java™ Wireless Toolkit for CLDC 包含一個名稱爲 `_main.mks` 的 ME 金鑰庫，位於 `appdb` 子目錄中。此金鑰庫包含預設 Java SE 平台金鑰庫中存在的所有憑證（此金鑰庫隨附於 Java SE SDK 的安裝中）。

---

## B.6 使用 Stub 產生器

行動用戶端可以使用 Stub 產生器存取 Web 服務。`wscompile` 工具會產生在 XML (JAX) RPC 用戶端與服務中使用的 Stub、Tie、Serializer 與 WSDL 檔案。此工具會讀取配置檔案，該檔案可指定 WSDL 檔案、模型檔案或編譯的服務端點介面。Stub 產生器的指令語法如下：

```
wscompile [options] configuration-files
```

### B.6.1 選項

表格 B-3 `wscompile` 指令的選項

選項	描述
<code>-d output directory</code>	指定放置所產生輸出檔案的位置
<code>-f:features</code>	啟用指定功能
<code>-features:features</code>	與 <code>-f:features</code> 相同
<code>-g</code>	產生除錯資訊
<code>-gen</code>	與 <code>-gen:client</code> 相同
<code>-gen:client</code>	產生用戶端工件 (Stub 等)
<code>-httpproxy:host:port</code>	指定 HTTP 代理伺服器（預設連接埠爲 8080）
<code>-import</code>	僅產生介面與值類型
<code>-model file</code>	寫入內部模型至指定的檔案

**表格 B-3** wscompile 指令的選項 (繼續)

選項	描述
-O	最佳化產生的程式碼
-s <i>directory</i>	指定放置所產生原始碼檔案的位置
-verbose	輸出關於編譯器處理內容的訊息
-version	顯示版本資訊
-cldc1.0	設定 CLDC 版本為 1.0 (預設值)。浮點數及雙精度浮點數變為 String。
-cldc1.1	設定 CLDC 版本為 1.1 (可使用浮點數及雙精度浮點數)
-cldc1.0info	顯示所有的 CLDC 1.0 資訊和警告訊息。

**備註** – 務必指定一個 -gen 選項。-f 選項需要以逗號分隔的功能清單。

表格 B-4 列出可以跟在 -f 選項後面的功能 (以逗號分隔)。wscompile 工具會讀取 WSDL 檔案、編譯的服務端點介面 (SEI) 或模型檔案做為輸入。[ 檔案類型 ] 欄說明哪些檔案可以與特定功能共同使用。

**表格 B-4** wscompile 的受支援指令功能 (-f)

選項	描述	檔案類型
explicitcontext	開啓明確的服務環境的對映	WSDL
nodatabinding	關閉文字列編碼資料連結	WSDL
noencodetypes	關閉編碼類型資訊	WSDL、SEI、模型
nomultirefs	關閉多重參照支援	WSDL、SEI、模型
novalidation	關閉匯入的 WSDL 文件完全驗證	WSDL
searchschema	積極搜尋子類型模式	WSDL
serializeinterfaces	開啓介面類型直接序列化	WSDL、SEI、模型
wsi	啓用 WSI-Basic Profile 功能 (預設)	
resolveidref	解析 xsd:IDREF	
nounwrap	取消不換行	

#### 範例

```
wscompile -gen -d generated config.xml
wscompile -gen -f:nounwrap -O -cldc1.1 -d generated config.xml
```



# 本土化

---

本附錄說明如何設定 Sun Java™ Wireless Toolkit for CLDC 中顯示的語言，以及模擬環境的本土化設定。

---

## C.1 語言環境設定

語言環境是指具有相同語言、習俗或文化傳統的地理或政治區域或團體。在軟體中，語言環境是指包含必要資訊，使軟體適用於特定地區的檔案、資料與程式碼的集合。

有些軟體使用語言環境，以針對使用者設定合適的資訊，例如：

- 對使用者顯示的訊息
- 使用的字型或其他與書寫相關的資訊

依預設，使用者介面中的所有字串，都將以支援平台的語言環境的語言顯示。

例如，在執行日文版 Microsoft Windows 的機器上，只要已下載並安裝了已正確本土化的 Sun Java™ Wireless Toolkit for CLDC，工具組便可以顯示日文字元。

您可以設定 `wtk.locale` 特性，讓使用者介面以指定語言環境的語言顯示。如第 3-13 頁的第 3.8 節「配置 Wireless Toolkit」中所討論，您必須將 `kttools.properties` 從安裝目錄複製到您的工作目錄 (`workdir/wtklib/kttools.properties`)，然後編輯您的副本。例如，當您想在日文版的機器上執行工具組，但仍要讓使用者介面以英文顯示時，可將語言環境特性設定為 `en-US`，並確定已下載並安裝了適當的 Sun Java™ Wireless Toolkit for CLDC 增補程式。

---

## C.2 模擬語言環境

裝置的語言環境包含在系統特性 `microedition.locale` 中。您可以選擇 [編輯] > [偏好]，然後選取 [國際化] 來變更模擬器的語言環境。請從組合方塊中選擇一種語言環境或直接鍵入該語言環境。

如需關於 `microedition.locale` 的資訊，請參閱 MIDP 規格。

---

## C.3 字元編碼

CLDC 系統特性 `microedition.encoding` 定義 MIDP 環境的預設字元編碼名稱。在 Sun Java™ Wireless Toolkit for CLDC 模擬器中，此特性是根據所使用的基礎視窗系統來設定。此特性值是設為在同一個視窗系統上執行的 Java SE 平台預設編碼。例如，在英文版視窗系統中，其編碼設定如下：

```
microedition.encoding=ISO8859_1
```

您可以將 `microedition.encoding` 特性加入 `workdir\wtklib\ktools.properties` 檔中，即可置換預設值。例如，如果您想將 UTF-8 當作 Microsoft Windows 的預設設定，可將 `workdir\wtklib\ktools.properties` 檔中的特性設定如下：

```
microedition.encoding=UTF-8
```

如需字元編碼的詳細資訊，請參閱 CLDC 規格。

---

**備註** – 所有 Java SE 平台編碼器皆可於模擬環境中使用。如需瞭解如何限制指定裝置的可用編碼器清單，請參閱「Sun Java™ Wireless Toolkit for CLDC 基本自訂指南」。

---

---

## C.4 Java 技術編譯器編碼設定

`javac.encoding` 特性會決定 `javac` 編譯器用來編譯原始檔的編碼方式。此特性值是設為在同一個視窗系統上執行的 Java SE 平台預設編碼。

將 `javac.encoding` 特性加入 `ktools.properties` 檔案中即可置換預設值。例如，如果您使用的是英文系統，但需要編譯日文資源束時，您可以指定日文版的字元集，例如：

```
javac.encoding=EUCJIS
```

---

## C.5 預設模擬器的字型支援

用於模擬環境中的預設字型，是根據基礎視窗系統語言環境來設定的。依預設，MIDP 環境字型會對映到預設 Java SE 平台的 Java 技術字型。這些字型通常支援目前視窗語言環境所需的所有字元。

您可以置換這些字型，以支援預設字型所不支援的其他字元。如需相關配置資訊，請參閱「Sun Java Wireless Toolkit for CLDC 基本自訂指南」。



# 索引

---

## 符號

- classpath 選項, B-2
- help 選項, B-4
- import 指令, B-10
- version 選項, B-4
- Xdebug 選項, B-5
- Xquery 選項, B-5
- Xrunjdpw 選項, B-5
- Xverbose 選項, B-5

## 英文

- AMR, 8-2
- AMS, 2-8
- CBS 訊息, 傳送, 7-6
- cref, 14-1
- emulator 指令, B-4
- URLConnection API, 10-1
- URLConnection API 中的根目錄, 10-2
- in.use 檔案, 4-5
- IrDA, 11-1
- J2ME Web Services Specification, 12-1
- JAD 檔案, 2-8
  - MIME 類型, 2-15
  - 建立, 2-8
  - 屬性, 3-3
- JadTool, B-9

## JAR 檔案

- MIME 類型, 2-15
- 在封裝中, 2-8
- 建立, 2-8

## Java Cryptography Extension (JCE) 金鑰庫, B-10

- JSR 75, 10-1
- JSR 82, 11-1, A-5
- JSR 118, 6-1
- JSR 120, 7-1
- JSR 135, 8-1
- JSR 172, 12-1, A-28
- JSR 177, 14-1
- JSR 179, 13-1, A-10
- JSR 180, 15-1, A-20, A-43
- JSR 184, 9-1, A-17
- JSR 185, 6-3
- JSR 205, 7-1
- JSR 211, A-7
- JSR 226, 9-3, A-45
- JSR 229, 16-1, A-25
- JSR 234, A-4
- JSR 238, 17-1, A-22
- JSR 239, 9-4
- JSR 248, 6-3
- JSR 75, A-36, A-38
- JTWI 保護網域, 6-4
- keytool 公用程式, B-10

- ktools.properties, C-1
- ktools.properties, 3-14
- M3G, 9-1
- MEKeyTool, B-10
- MIA, 17-1
- microedition.encoding 特性, C-2
- MIDlet
  - JAR 檔案, 2-8
  - 修改, 3-5
  - 描述元, 2-8
  - 增加新的, 3-5
- MIDlet 套件, 使用實際金鑰簽署, 6-6
- MIDlet 套件, 簽署, 6-4
- MIME 類型, 2-15
- MMAPI, 8-1, A-28
- Mobile 3D Graphics API, 9-1
- Mobile Internationalization API, 17-1
- Mobile Media API, 8-1
- Mobile Media API (MMAPI), 8-1
  - 格式與協定, 8-1
  - 擷取, 8-4
- MSA 保護網域, 6-3
- OBEX, 11-1
  - 示範, A-35
  - 偏好, 11-3
- OpenGL(R) ES, 9-4
- Payment API, 16-1, A-25
- PDA Optional Packages, 10-1
- PDAP, 10-1
- PIM API, 10-2
- RevisionControl 特性, 3-14
- run 選項, B-4
- SATSA, 14-1
- SATSA 示範, A-40
- SIP API, 15-1
- SMS 二進位訊息, 傳送, 7-5
- SMS 文字訊息, 傳送, 7-4
- SVG, 9-3
- SVG 描繪, 9-3
- toolkit, 1-1

- USB 加密碟, 6-9
- Web Services Specification, 12-1
- Web 服務, Stub 產生器, 12-1
- Web 服務的 Stub 產生器, 12-1
- Wireless Messaging API, 7-1
- Wireless Toolkit
  - 從指令行執行, B-1
  - 憑證管理員公用程式, B-1
- WMA, 7-1
- WMA 主控台, 7-3, A-52
- wscompile 工具, B-11
- WSDL 檔案, 12-2

## 三畫

- 工具組
  - 啟動, 1-4, 2-1
  - 應用程式目錄, 3-14

## 四畫

- 支援的 API, 1-7
- 方法效能評測, 5-1

## 五畫

- 代理伺服器, 4-3
- 永久性儲存裝置, 4-4, 4-5
  - 清除資料庫, 4-5
- 生成 (指令行), B-2
- 生成 (透過使用者介面), 2-5
- 由遠端部署的應用程式, 2-15
- 目標平台, 3-2
- 示範, A-1
  - 原始碼, 1-2

## 六畫

- 在 Web 伺服器上部署, 2-15
- 字元編碼, C-2
- 字型支援, C-3

## 七畫

- 位置 API, 13-1
- 更新, 1-6

## 八畫

- 版本控制， 3-14
- 金鑰庫, JCE， B-10
- 金鑰組
  - 建立， 6-5
  - 匯入， 6-7
- 金鑰管理， 6-5

## 九畫

- 保護網域， 6-1
  - JTWI， 6-4
  - MSA， 6-3
- 封裝
  - 指令行範例， B-3
- 指令行作業， B-1
- 指令路徑， B-1

## 十畫

- 個人資訊管理 (PIM) API， 10-1
- 修正控制， 3-14
- 修正控制系統 (RCS)， 3-14
- 修正控制檔案， 3-14
- 原始碼
  - 位置， 2-2
  - 建立， 2-4
- 效能， 4-5
- 效能評測器， 5-1
  - 調用圖， 5-3
- 記憶體監視器， 5-4
  - 物件詳細資訊， 5-5
  - 圖形， 5-4
- 訊息 URL http
  - //www.ietf.org/rfc/rfc3267.txt， 8-2
- 訊息傳送, 網路模擬， 7-3
- 訊息樹狀結構排序， 5-8
- 追蹤選項， B-5
- 除錯， 2-15
  - 從指令行， B-5
  - 選項， B-5

## 十一畫

- 堆疊， 4-5
- 堆疊大小， 4-4
- 專案， 2-1， 3-1
  - MIDlet， 3-5
  - 生成， 2-5
  - 封裝， 2-8
  - 建立， 2-2
  - 原始碼， 2-2
  - 執行， 2-6
  - 從 JAD/JAR 建立， 2-13
  - 推送註冊表， 3-6
  - 部署在實際裝置上， 2-12
  - 程式庫， 3-11
  - 選取 API， 3-1
  - 屬性， 3-3
- 從指令行產生 Stub， B-11
- 從指令行管理憑證， B-10
- 推送註冊表， 3-6
- 清除 RMS， 3-11
- 清單檔案, 建立， B-3
- 混淆器， 2-14
  - 安裝 ProGuard， 2-14

## 十二畫

- 描述元， 2-8
  - 屬性， 3-3
- 程式庫， 3-11
- 開發周期
  - 完整， 2-7
  - 簡單， 2-3

## 十三畫

- 經由 OTA 執行， 2-8， 6-3
- 鈴聲， 8-4
- 電話號碼, 在模擬器中設定， 7-1
- 預驗證， 2-5， B-2
  - 指令行範例， B-2

## 十四畫

網路監視器， 5-7  
    排序訊息， 5-8  
    與 WMA 一起使用， 7-8  
    篩選， 5-8  
語言環境， C-1

## 十五畫

暫停與重新繼續， 4-7  
模擬器， 4-1  
    面板， 4-1  
    效能， 4-5  
    偏好， 4-3， B-6  
    單獨執行， 4-7  
    預設字型支援， C-3  
    預設保護網域， B-9  
    語言支援， C-1  
    語言環境， C-2  
    鍵盤捷徑， 4-3  
編碼， javac， 3-14  
編譯器編碼方式， C-2  
調用圖， 5-3

## 十六畫

憑證匯入， 6-8  
憑證管理， 6-7  
憑證管理員公用程式， B-1  
選擇性的 API， 1-7

## 十七畫

儲存偏好， 4-4， 4-5  
應用程式  
    在 Web 伺服器上部署， 2-15  
    遠端執行， 2-15  
應用程式目錄，設定， 3-14  
應用程式描述元， 2-8  
檢查更新， 1-6

## 十八畫

瀏覽器  
    於配置檔案中指定， B-5  
    於模擬器執行期間指定， B-4  
藍芽， 11-1

## 十九畫

簽署 MIDlet 套件， 6-4， B-9  
簽署的 MIDlet 套件， 6-1

## 二十畫以上

屬性， 3-3  
權限， 6-1