



用户指南

Sun JavaTM Wireless Toolkit for CLDC

2.5.2 版

Sun Microsystems, Inc.
www.sun.com

2007 年 9 月

版权所有 ©2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文档中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家 / 地区申请的一项或多项其他专利或待批专利。

美国政府权利 — 商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR (Federal Acquisition Regulations, 即“联邦政府采购法规”) 的适用条款及其补充条款。必须依据许可证条款使用。本发行版可能包含由第三方开发的内容。

Sun、Sun Microsystems、Sun 徽标、Java、Javadoc、Java Community Process、JCP、JDK、JRE、J2ME 和 J2SE 是 Sun Microsystems, Inc. 在美国和其他国家 / 地区的商标或注册商标。

OpenGL 是 Silicon Graphics, Inc. 的注册商标。

本服务手册所介绍的产品以及所包含的信息受美国出口控制法制约，并应遵守其他国家 / 地区的进出口法律。严禁将本产品直接或间接地用于核设施、导弹、生化武器或海上核设施，也不能直接或间接地出口给核设施、导弹、生化武器或海上核设施的最终用户。严禁出口或转口到美国禁运的国家 / 地区以及美国禁止出口清单中所包含的实体，包括但不限于被禁止的个人以及特别指定的国家 / 地区的公民。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性和非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。

目录

前言 xiii

1. 简介 1-1

- 1.1 关于多用户环境 1-1
 - 1.1.1 安装目录和工作目录 1-1
 - 1.1.2 工作目录文件 1-2
- 1.2 快速入门 1-4
- 1.3 Toolkit 组件 1-5
- 1.4 Toolkit 功能 1-6
- 1.5 工具包更新 1-6
- 1.6 支持的技术 1-7

2. 开发 MIDlet 套件 2-1

- 2.1 关于项目 2-1
- 2.2 简单开发周期 2-3
 - 2.2.1 编辑源代码 2-4
 - 2.2.2 生成 2-5
 - 2.2.3 运行 2-6
- 2.3 完整开发周期 2-7
 - 2.3.1 打包 2-8

| | | |
|-------|------------------|------|
| 2.3.2 | 安装 | 2-8 |
| 2.3.3 | 运行 | 2-13 |
| 2.4 | 从 MIDlet 套件创建项目 | 2-13 |
| 2.5 | 使用混淆器 | 2-14 |
| 2.5.1 | 安装 ProGuard | 2-14 |
| 2.5.2 | 使用 ProGuard | 2-14 |
| 2.6 | 使用调试器 | 2-15 |
| 2.7 | 在 Web 服务器上部署应用程序 | 2-16 |

3. 使用项目 3-1

| | | |
|-------|---------------------|------|
| 3.1 | 选择 API | 3-1 |
| 3.2 | 更改 MIDlet 套件属性 | 3-3 |
| 3.3 | 操纵 MIDlet | 3-5 |
| 3.4 | 使用推送注册表 | 3-6 |
| 3.5 | 设置内容处理器 | 3-7 |
| 3.6 | 项目目录结构 | 3-11 |
| 3.7 | 使用第三方库 | 3-11 |
| 3.7.1 | 使用外部 API | 3-12 |
| 3.7.2 | 用于一个项目的第三方库 | 3-13 |
| 3.7.3 | 用于所有项目的第三方库 | 3-13 |
| 3.8 | 配置 Wireless Toolkit | 3-13 |
| 3.8.1 | 更改控制台字体 | 3-14 |
| 3.8.2 | 设置应用程序目录 | 3-14 |
| 3.8.3 | 设置 javac 编码属性 | 3-14 |
| 3.8.4 | 使用修订控制系统 | 3-14 |

4. 使用仿真器 4-1

| | | |
|-----|-------|-----|
| 4.1 | 仿真器样机 | 4-1 |
| 4.2 | 仿真器控制 | 4-2 |

| | | |
|---------|----------|-----|
| 4.3 | 设置仿真器首选项 | 4-3 |
| 4.3.1 | 网络代理 | 4-3 |
| 4.3.2 | 存储大小 | 4-4 |
| 4.3.2.1 | 永久存储 | 4-4 |
| 4.3.2.2 | 堆栈大小 | 4-5 |
| 4.3.3 | 调整仿真器性能 | 4-5 |
| 4.4 | 暂停和恢复 | 4-7 |
| 4.5 | 单独运行仿真器 | 4-7 |
| 4.6 | 使用第三方仿真器 | 4-8 |

5. 监视应用程序 5-1

| | | |
|-------|--------------|-----|
| 5.1 | 使用事件探查器 | 5-1 |
| 5.1.1 | 方法调用图 | 5-3 |
| 5.1.2 | 执行时间和调用次数 | 5-4 |
| 5.1.3 | 保存和装入事件探查器信息 | 5-4 |
| 5.2 | 使用内存监视器 | 5-4 |
| 5.2.1 | 保存和装入内存监视器信息 | 5-6 |
| 5.3 | 使用网络监视器 | 5-7 |
| 5.3.1 | 过滤消息 | 5-8 |
| 5.3.2 | 对消息进行排序 | 5-8 |
| 5.3.3 | 保存和装入网络监视器信息 | 5-9 |
| 5.3.4 | 清除消息树 | 5-9 |

6. 安全和 MIDlet 签名 6-1

| | | |
|-------|--|-----|
| 6.1 | 权限 | 6-1 |
| 6.2 | 选择安全策略 | 6-3 |
| 6.2.1 | MSA 保护域 | 6-3 |
| 6.2.2 | Java for the Wireless Toolkit Industry 保护域 | 6-4 |
| 6.3 | 签署 MIDlet 套件 | 6-4 |

| | | |
|-------|---------------|------|
| 6.4 | 管理密钥 | 6–5 |
| 6.4.1 | 创建新密钥对 | 6–5 |
| 6.4.2 | 获得真实密钥 | 6–6 |
| 6.4.3 | 导入现有密钥对 | 6–7 |
| 6.4.4 | 删除密钥对 | 6–7 |
| 6.5 | 管理证书 | 6–7 |
| 6.5.1 | 启用和禁用证书 | 6–8 |
| 6.5.2 | 导入证书 | 6–8 |
| 6.5.3 | 删除证书 | 6–8 |
| 6.6 | USB 令牌支持 | 6–9 |
| 6.6.1 | 安装 USB 令牌驱动程序 | 6–9 |
| 6.6.2 | 使用 USB 令牌 | 6–10 |

7. 使用 Wireless Messaging API 7–1

| | | |
|-------|-----------------|-----|
| 7.1 | 设置仿真器电话号码 | 7–1 |
| 7.2 | 模拟不可靠的网络 | 7–3 |
| 7.3 | 通过 WMA 控制台发送消息 | 7–3 |
| 7.3.1 | 发送文本 SMS 消息 | 7–4 |
| 7.3.2 | 发送二进制 SMS 消息 | 7–5 |
| 7.3.3 | 发送文本或二进制 CBS 消息 | 7–6 |
| 7.3.4 | 发送 MMS 消息 | 7–7 |
| 7.4 | 在 WMA 控制台中接收消息 | 7–8 |
| 7.5 | 结合使用网络监视器与 WMA | 7–8 |

8. 使用 Mobile Media API 8–1

| | | |
|-------|--------------------------------------|-----|
| 8.1 | 支持的格式和协议 | 8–1 |
| 8.2 | 自适应多速率 (Adaptive Multi-Rate, AMR) 内容 | 8–2 |
| 8.2.1 | Windows | 8–2 |
| 8.2.2 | Linux | 8–2 |

| | | |
|---|---------------------------|------|
| 8.2.2.1 | 启用 AMR 支持 | 8-2 |
| 8.2.2.2 | AMR 格式支持 | 8-3 |
| 8.3 | 使用 MediaControlSkin | 8-3 |
| 8.4 | 媒体捕获 | 8-4 |
| 8.5 | 正常的 MIDlet | 8-4 |
| 8.6 | 铃声 | 8-4 |
| 8.6.1 | 下载铃声 | 8-4 |
| 8.6.2 | 铃声格式 | 8-5 |
| 9. 使用 Mobile Graphics 9-1 | | |
| 9.1 | 使用 Mobile 3D Graphics API | 9-1 |
| 9.1.1 | 立刻模式 | 9-1 |
| 9.1.2 | 保留模式 | 9-2 |
| 9.1.3 | 以质量博取速度 | 9-2 |
| 9.1.4 | 创建 Mobile 3D 图形内容 | 9-2 |
| 9.2 | 渲染可伸缩矢量图形内容 | 9-3 |
| 9.3 | OpenGL® ES 概述 | 9-3 |
| 10. 使用 PIM 和 FileConnection API 10-1 | | |
| 10.1 | FileConnection API | 10-1 |
| 10.2 | PIM API | 10-3 |
| 11. 使用蓝牙和 OBEX API 11-1 | | |
| 11.1 | 蓝牙模拟环境 | 11-1 |
| 11.2 | OBEX 红外传输 | 11-1 |
| 11.3 | 设置 OBEX 和蓝牙首选项 | 11-2 |
| 11.3.1 | OBEX 首选项 | 11-3 |
| 11.3.2 | 蓝牙内部特性 | 11-3 |
| 11.3.3 | 蓝牙系统特性 | 11-3 |
| 11.3.4 | 蓝牙 BCC 特性 | 11-4 |

- 12. 使用 Web 服务 12-1**
- 13. 使用 Location API 13-1**
 - 13.1 设置仿真器运行时的位置 13-1
 - 13.2 配置位置提供者 13-4
 - 13.3 设置路标 13-5
- 14. 使用 SATSA 14-1**
 - 14.1 仿真器中的卡插槽 14-2
 - 14.2 使用 Java Card 平台模拟器 14-3
 - 14.3 使用具有 SATSA 的网络监视器 14-3
 - 14.4 调整访问控制 14-4
 - 14.4.1 指定 PIN 属性 14-5
 - 14.4.2 指定应用程序权限 14-5
 - 14.4.3 访问控制文件示例 14-7
- 15. 使用 SIP 15-1**
 - 15.1 了解注册器和代理 15-1
 - 15.2 SIP 设置 15-2
 - 15.3 网络监视器中的 SIP 通信流量 15-3
 - 15.4 SIP 代理服务器和注册器 15-4
- 16. 使用 Payment API 16-1**
 - 16.1 项目付费设置 16-1
 - 16.2 直接编辑付费属性 16-4
 - 16.3 付费首选项 16-4
 - 16.4 查看事务历史 16-6
 - 16.5 监视付费 16-7
- 17. 使用 Mobile Internationalization API 17-1**
 - 17.1 设置仿真器的语言环境 17-1

- 17.2 查看应用程序资源 17-1
- 17.3 使用语言环境 17-3
- 17.4 使用资源文件 17-3
- 17.5 使用资源 17-4

A. 演示应用程序 A-1

- A.1 概述 A-1
- A.2 一般说明 A-3
- A.3 Advanced Multimedia Supplements A-4
- A.4 Bluetooth Demo A-6
- A.5 CHAPIDemo A-7
- A.6 CityGuide A-11
- A.7 演示 A-14
 - A.7.1 颜色 A-14
 - A.7.2 属性 A-14
 - A.7.3 Http A-15
 - A.7.4 FontTestlet A-16
 - A.7.5 股票行情 A-16
 - A.7.5.1 使用设置 A-17
 - A.7.5.2 股票跟踪器 A-17
 - A.7.5.3 What If? A-17
 - A.7.5.4 Alerts A-18
 - A.7.6 证明书 A-18
 - A.7.7 ManyBalls A-19
- A.8 Demo3D A-19
 - A.8.1 Life3D A-19
 - A.8.2 PogoRoo A-21
 - A.8.3 retainedmode A-22
- A.9 GoSIP A-22

- A.10 i18nDemo A-24
- A.11 JBricks A-27
- A.12 JSR172Demo A-30
- A.13 MobileMediaAPI A-30
 - A.13.1 Simple Tones A-30
 - A.13.2 Simple Player A-31
 - A.13.3 Video A-33
 - A.13.4 暂停音频测试 A-33
 - A.13.5 MobileMediaAPI 的属性 A-34
- A.14 网络演示 A-34
 - A.14.1 套接字演示 A-34
 - A.14.2 数据报演示 A-36
- A.15 ObexDemo A-37
- A.16 PDAPDemo A-38
 - A.16.1 浏览文件 A-38
 - A.16.2 PIM API A-41
- A.17 SATSADemos A-43
 - A.17.1 APDUMIDlet A-44
 - A.17.2 SATMIDlet A-44
 - A.17.3 CryptoMIDlet A-44
 - A.17.4 MohairMIDlet A-45
- A.18 SATSAJCRMIDemo A-46
- A.19 SIPDemo A-46
- A.20 SVGContactList A-46
- A.21 SVGDemo A-48
 - A.21.1 SVG 浏览器 A-48
 - A.21.2 渲染 SVG 图像 A-48
 - A.21.3 播放 SVG 动画 A-48

| | | |
|--------|---------------|------|
| A.21.4 | 从头开始创建 SVG 图像 | A-49 |
| A.21.5 | 弹力球 | A-49 |
| A.21.6 | 优化的菜单 | A-49 |
| A.21.7 | 图像装饰器 | A-50 |
| A.21.8 | 定位服务 | A-52 |
| A.22 | WMADemo | A-53 |

B. 命令行参考 **B-1**

| | | |
|-------|--------------|------|
| B.1 | 先决条件 | B-1 |
| B.2 | 开发周期 | B-2 |
| B.2.1 | 生成 | B-2 |
| B.2.2 | 打包 | B-3 |
| B.2.3 | 运行 | B-4 |
| B.2.4 | 调试 | B-6 |
| B.3 | 启动工具包 GUI 组件 | B-6 |
| B.4 | 设置仿真器首选项 | B-7 |
| B.5 | 使用安全功能 | B-9 |
| B.5.1 | 更改仿真器的默认保护域 | B-9 |
| B.5.2 | 签名 MIDlet 套件 | B-9 |
| B.5.3 | 管理证书 | B-11 |
| B.6 | 使用桩模块生成器 | B-12 |
| B.6.1 | 选项 | B-12 |

C. 本地化 **C-1**

| | | |
|-----|----------------|-----|
| C.1 | 语言环境设置 | C-1 |
| C.2 | 仿真的语言环境 | C-2 |
| C.3 | 字符编码 | C-2 |
| C.4 | Java 技术编译器编码设置 | C-2 |
| C.5 | 默认仿真器中的字体支持 | C-3 |

前言

本文档介绍如何使用 Sun Java Wireless Toolkit for CLDC。

目标读者

本指南供使用 Sun Java™ Wireless Toolkit for CLDC 创建 Mobile Information Device Profile (MIDP) 应用程序的开发人员使用。本书并非 MIDP 编程教程，也不是工具包支持的任何附加 API 的编程教程。您应当已经知道如何使用 Mobile Information Device Profile (MIDP) 和 Connected Limited Device Configuration (CLDC)。

如果您需要 Java 编程语言方面的快速入门帮助，请尝试访问 New to Java Center:

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

要快速学习 MIDP 编程，请阅读 Learning Path: Getting Started with MIDP 2.0:

<http://developers.sun.com/techtopics/mobile/learn/midp/midp20/>

相关文档

本节列出相关的 Java Platform, Micro Edition (Java ME) 规范。Java ME 以前称为 Java 2 Platform, Micro Edition 或 J2ME, 如您在某些规范名称中所见。尽管这些规范具有权威性, 但不一定最易于理解。有关针对开发人员的文章, 请访问 Sun 的 Mobility Web 站点:

<http://developers.sun.com/techtopics/mobile/>

表 P-1 相关文档

| 主题 | 书名 |
|---|--|
| 自定义 Sun Java™ Wireless Toolkit for CLDC | Sun Java™ Wireless Toolkit for CLDC 基本自定义指南 |
| 发行说明 | Sun Java™ Wireless Toolkit for CLDC 发行说明 |
| 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 |
| Bluetooth and OBEX - JSR 82 | Java APIs for Bluetooth |
| MIDP 2.1 - JSR 118 | Mobile Information Device Profile 2.0 (最终版本 2 指 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 |

本书的结构

本指南包含以下章节和附录：

[第 1 章](#) 介绍 Sun Java™ Wireless Toolkit for CLDC 及其提供的开发功能。

[第 2 章](#) 介绍创建和运行 MIDlet 的开发过程。

[第 3 章](#) 介绍如何使用该工具包中的项目。您将了解如何调整项目属性、操纵 MIDlet 以及如何使用推送注册表，并了解项目的目录结构。

[第 4 章](#) 介绍仿真器，并说明如何调整仿真器选项以及利用它的诸多功能。

[第 5 章](#) 介绍如何使用方法事件探查器、内存监视器以及网络监视器检查应用程序的性能。

[第 6 章](#) 介绍如何签名 MIDlet 套件，以及管理密钥和证书。

[第 7 章](#) 详细介绍对运行和测试无线消息传送应用程序的支持。

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

[第 9 章](#) 提供有关开发 3D 图形内容的信息。

[第 10 章](#) 介绍工具包如何实现访问本地文件和个人信息，如联系人信息和约会日程。

[第 11 章](#) 介绍工具包的蓝牙和 OBEX 仿真环境。

[第 12 章](#) 介绍如何使用 Web 服务桩模块生成器。

[第 13 章](#) 介绍如何使用仿真器的位置功能。

[第 14 章](#) 介绍工具包对 SATSA 的支持。

[第 15 章](#) 介绍工具包的 SIP 支持。

[第 16 章](#) 介绍工具包的 Payment API 功能。

[第 17 章](#) 介绍如何管理 Mobile Internationalization API 的资源。

[附录 A](#) 介绍 Sun Java™ Wireless Toolkit for CLDC 中包含的演示应用程序。

[附录 B](#) 介绍如何通过命令行使用 Sun Java™ Wireless Toolkit for CLDC 的功能。

[附录 C](#) 介绍 Sun Java™ Wireless Toolkit for CLDC 中的本地化功能。

印刷约定

表 P-2 介绍在本文档中的字体使用情况。

表 P-2 印刷约定

| 字体 | 含义 | 示例 |
|----------------------------|---|---|
| AaBbCc123 (等宽) | 命令、文件和目录的名称；计算机屏幕输出 | 编辑 .login 文件。 使用 ls -a 列出所有文件。 % You have mail. |
| AaBbCc123 (等宽粗体) | 用户键入的内容，与计算机屏幕输出的显示不同。 | % su Password: |
| <i>AaBbCc123</i> (斜体) | 保留未译的新词或术语以及要强调的词。 将用实际名称或值替换的命令或路径名中的占位符。 | 这些被称为 <i>class</i> 选项。 该文件位于 <i>install-dir/bin</i> 目录中。 |
| 新词术语强调 | 新词或术语以及要强调的词 | 您必须成为超级用户才能执行此操作。 |
| 《书名》 | 书名 | 阅读《用户指南》的第 6 章。 |
| {AaBbCc.dir} | 可变的文件名和目录。 | 本书中， <i>toolkit</i> 始终指 Sun JavaTM Wireless Toolkit for CLDC 的安装目录。 |

联机访问 Sun 文档

下列站点提供与 Java 技术有关的技术文档：

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

欢迎您提出意见

我们致力于提高文档的质量。请通过 developers.sun.com 使用电子邮件提交您的意见和建议。

简介

本书介绍如何使用 Sun Java™ Wireless Toolkit for CLDC。

Sun Java™ Wireless Toolkit for CLDC 是一个工具集，通过它可以创建用于移动电话和其他无线设备的应用程序。虽然 Sun Java™ Wireless Toolkit for CLDC 建立在 Mobile Information Device Profile (MIDP) 2.1 基础之上，但它还支持许多可选包，是一个功能强大的开发工具包。

1.1 关于多用户环境

Sun Java Wireless Toolkit 2.5.2 for CLDC 可以安装在运行所支持的 Windows 或 Linux 版本的系统上。在主机上具有帐户的所有用户均可以单独访问此工具包，也可以同时访问此工具包。

1.1.1 安装目录和工作目录

Windows 路径中包含驱动器号，并将反斜杠用作目录分隔符。Linux 路径使用正斜杠。在 Linux 路径中，~ 表示 Linux 用户的主目录。

为支持多个用户，工具包将创建安装目录来作为用于复制的源使用。本文档使用变量 *workdir* 表示工具包工作目录，使用 *toolkit* 表示安装目录。每个用户的个人文件都保存在名为 j2mewtk 的单独工作目录（对于所安装的每个版本，该目录都具有子目录）中。*workdir* 的默认位置通常为以下位置之一：

对于 Windows 系统： C:\Documents and Settings\User\j2mewtk\2.5.2
(其中 User 是您的帐户名)

对于 Linux 系统： ~/j2mewtk/2.5.2 (其中 ~ 是您的主目录)

本文档中介绍了在 Windows 和 Linux 之间存在的显著差异。如果仅涉及一种操作系统，您可以假定上述路径记法的差异适用。

1.1.2 工作目录文件

安装时，安装程序会将文件的子集从安装目录复制到每个用户的工作目录。工作目录的内容如下所示：

- **.settings**——最初包含 `security.properties`。
- **appdb**——会将整个应用程序数据库复制到您的工作目录。
- **apps**——会在您的工作目录中创建空的 `apps` 目录。如果打开演示项目，会将该项目自动复制到此位置。新建项目（[第 2.1 节](#)，[第 2-1 页上的“关于项目”](#)）时，新项目也会存储到此位置。
- **wtklib**——包含仿真器属性文件（请参见[第 4 章](#)）以及 HTTP 和 WMA 服务器和客户机的状态信息。
- **sessions**——这是监视工具（请参见[第 5 章](#)）的默认保存目标位置。最初该目录为空。

演示应用程序的所有源代码都可以在 `toolkit\apps` 中找到，并且每个演示都有自己的项目目录。放置在 `toolkit\apps` 中的所有应用程序对于所有用户都是可见的。

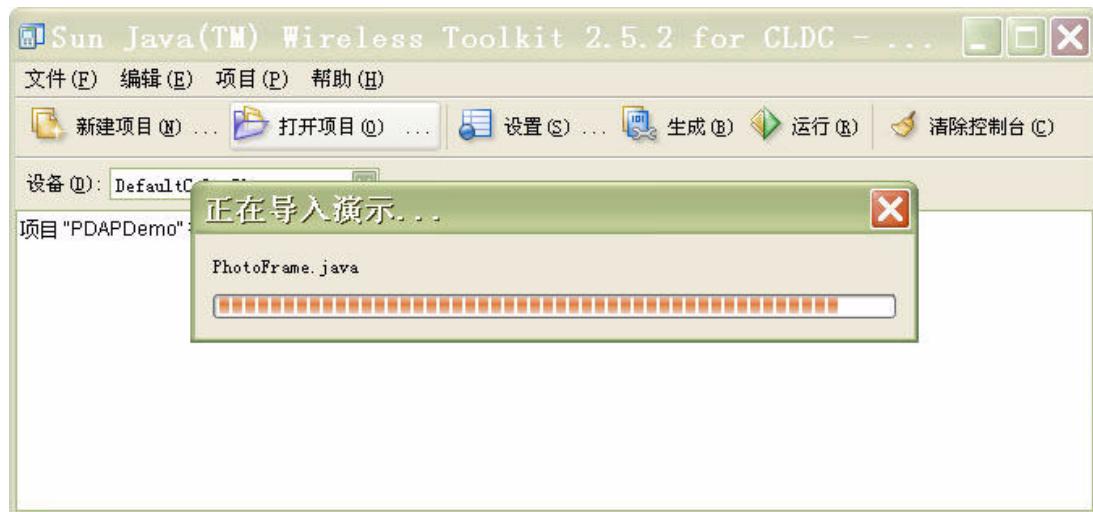
例如，在 Windows 上，MobileMediaAPI 演示的源代码位于 `toolkit\apps\MobileMediaAPI\src` 中。使用“打开项目”命令时，您可以看到位于安装目录和您工作目录中的项目（如果在这两个目录中都存在项目）。如图 1-1 中所示，本地目录中的项目以粗体显示，而安装目录中的项目以斜体显示。要仅显示您的本地文件，请清除显示可用演示复选框。

图 1-1 项目列表以粗体显示本地项目



当您打开安装目录中的项目时，会将应用程序的副本导入 `workdir\apps` 中。通常导入过程是觉察不到的，但在运行速度较慢的计算机上可能会看到进度对话框，如图 1-2 所示。任何自定义应仅针对您的本地副本进行。

图 1-2 将项目从 *toolkit* 导入 *workdir*



1.2 快速入门

如果您想快速入门，请尝试使用 Sun Java™ Wireless Toolkit for CLDC 中附带的演示应用程序。

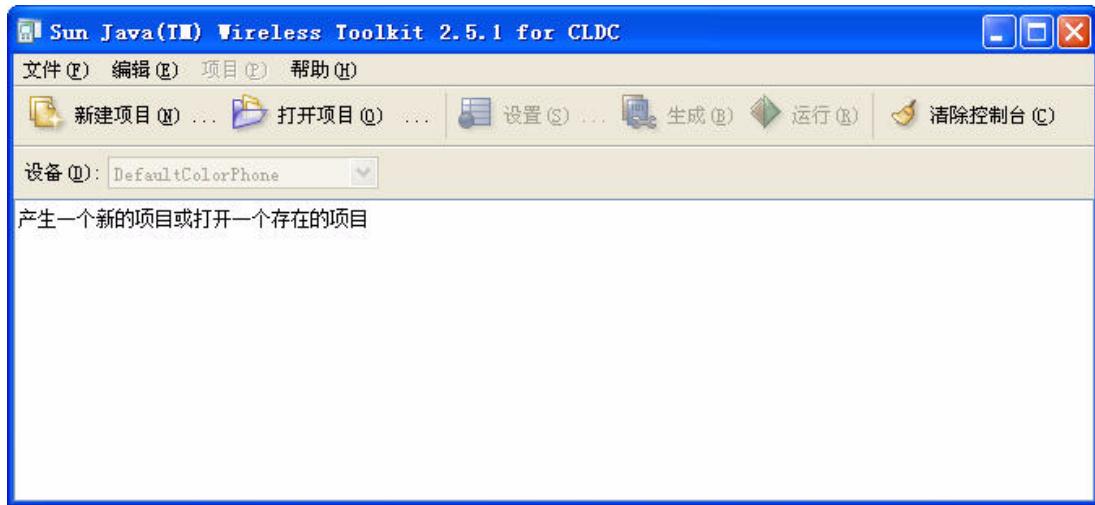
要运行演示，请按如下方式启动工具包。

- 在 Microsoft Windows 中，选择“开始”>“程序”>"Sun Java Wireless Toolkit 2.5.2 for CLDC" > "Wireless Toolkit 2.5.2".¹
- 在 Linux 上，将目录更改为 *toolkit/bin*。要运行工具包，请输入 `./ktoolbar`。

将打开类似于图 1-3 的窗口。

1. 根据 Microsoft Windows 的配置方式，可能需要选择“开始”>“所有程序”，而不是选择“开始”>“程序”。

图 1-3 工具包用户界面



单击“打开项目”按钮可以查看所有可用应用程序的列表。如图 1-1 中所示，斜体的名称表示存储在安装目录中的应用程序，而黑体的名称（如果有）表示存储在您工作目录中的应用程序。从中选择一个应用程序，然后单击对话框中的“打开项目”按钮。如果您之前未打开过该项目，则将在您的工作目录中创建一个副本，如第 1.1.2 节，第 1-2 页上的“[工作目录文件](#)”中所述。

打开应用程序后，请按“运行”按钮。仿真器将显示正在运行示例应用程序。

大多数演示很简单或自身带有说明，但有些演示有附加说明。对于某些演示，您需要使用“项目”>“通过 OTA 运行”，而不使用“运行”按钮。有关演示的一般说明和介绍，请参见[附录 A](#)。

1.3 Toolkit 组件

Sun Java™ Wireless Toolkit for CLDC 有三个主要组件：

- 用户界面可以自动执行创建 MIDP 应用程序的多项相关任务。
- 仿真器是一部模拟移动电话。可用于测试 MIDP 应用程序。
- 实用程序集合提供了其他有用的功能，包括文本消息传送控制台和加密实用程序。

通过用户界面您可以构建应用程序、启动仿真器以及启动实用程序。此外，仿真器和实用程序也能自行运行，这在很多情况下都很有用。例如，如果您想演示 MIDP 应用程序，则自行运行仿真器非常有用。

您唯一需要的另外一个工具是用于编辑源代码的文本编辑器。

1.4

Toolkit 功能

Sun Java™ Wireless Toolkit for CLDC 通过以下主要功能支持创建 MIDP 应用程序：

- **生成和打包：**您只需要编写源代码，其余的工作都可以交给工具包来完成。只需按下某个按钮，该工具包即可完成下列工作：编译源代码，预校验类文件以及对 MIDlet 套件打包。
- **运行和监视：**您可以在仿真器中直接运行某个 MIDlet 套件，也可以使用与在真实设备上安装应用程序类似的过程安装该套件。提供内存监视器、网络监视器和方法事件探查器，用于分析 MIDlet 的运行情况。
- **MIDlet 套件签名：**该工具包包含用于对 MIDlet 套件进行加密签名的工具。这对于在不同保护域中测试 MIDlet 的运行情况非常有用。

1.5

工具包更新

安装时，您可以选择激活“检查产品更新”功能。该功能使用网络每隔七天检查一次更新。要确定是否需要更新，它将收集以下信息：

- 操作系统
- 计算机上的“国家 / 地区和语言”设置
- 日期和当前使用的 Wireless Toolkit 的版本
- 由 Wireless Toolkit 生成的唯一随机用户 ID

这些数据用于改进产品。请阅读位于 <http://www.sun.com/privacy/index.html> 的 Sun 的隐私政策。

要激活或取消激活该功能，请选择“编辑”>“首选项”，然后单击“网络配置”。在面板的底部，选中或取消选中“检查更新”框。

1.6 支持的技术

Sun Java™ Wireless Toolkit for CLDC 支持通过 Java Community Process (JCP) 程序定义的多种标准应用程序编程接口 (Application Programming Interface, API)。表 1-1 显示了 API 及其规范的链接。

表 1-1 支持的 JCP 程序 API

| JSR API | 名称 URL |
|------------------------------|--|
| JSR 248 MSA 1.0 | Mobile Service Architecture http://jcp.org/en/jsr/detail?id=248 |
| JSR 185 JTWI 1.0 | Java Technology for the Wireless Industry http://jcp.org/en/jsr/detail?id=185 |
| JSR 139 CLDC 1.1 | Connected Limited Device Configuration http://jcp.org/en/jsr/detail?id=139 |
| JSR 118 MIDP 2.0 | Mobile Information Device Profile http://jcp.org/en/jsr/detail?id=118 |
| JSR 75 PIM 和文件 | PDA Optional Packages for the J2ME Platform http://jcp.org/en/jsr/detail?id=75 |
| JSR 82 Bluetooth and OBEX | Java APIs for Bluetooth http://jcp.org/en/jsr/detail?id=82 |
| JSR 135 MMAPI 1.1 | Mobile Media API http://jcp.org/en/jsr/detail?id=135 |
| JSR 172 | J2ME Web 服务规范 http://jcp.org/en/jsr/detail?id=172 |
| JSR 177 SATSA | Security and Trust Services API for Java ME http://jcp.org/en/jsr/detail?id=177 |
| JSR 179 位置 | Location API for Java ME http://jcp.org/en/jsr/detail?id=179 |
| JSR 180 SIP | SIP API for Java ME http://jcp.org/en/jsr/detail?id=180 |
| JSR 184 3D 图形 | Mobile 3D Graphics API for J2ME http://jcp.org/en/jsr/detail?id=184 |
| JSR 205 WMA 2.0 | Wireless Messaging API http://jcp.org/en/jsr/detail?id=205 |

表 1-1 支持的 JCP 程序 API (续)

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

开发 MIDlet 套件

本章介绍如何使用 Sun Java™ Wireless Toolkit for CLDC 来创建应用程序。首先介绍工具包项目，然后逐步完成开发过程。

创建 MIDlet 套件应用程序时，可能需要遵循两个基本开发周期。第一个周期较短，并且较为简单。在开发初期会很有用。第二个周期历时较长，但它为进行更全面、更真实的测试留下了余地。

在本章结尾处包含有关如何将 Sun Java™ Wireless Toolkit for CLDC 与混淆器和调试器这样的高级开发工具配合使用的信息。最后一节简单介绍如何配置 Web 服务器，以提供 MIDP 应用程序服务。

2.1 关于项目

在 Sun Java™ Wireless Toolkit for CLDC 中，各 MIDlet 套件被组织成一个个项目，其中一个项目的最终结果就是一个 MIDlet 套件。项目包含将用于生成 MIDlet 套件的所有文件，包括 Java 源文件、资源文件，以及 MIDlet 描述符。

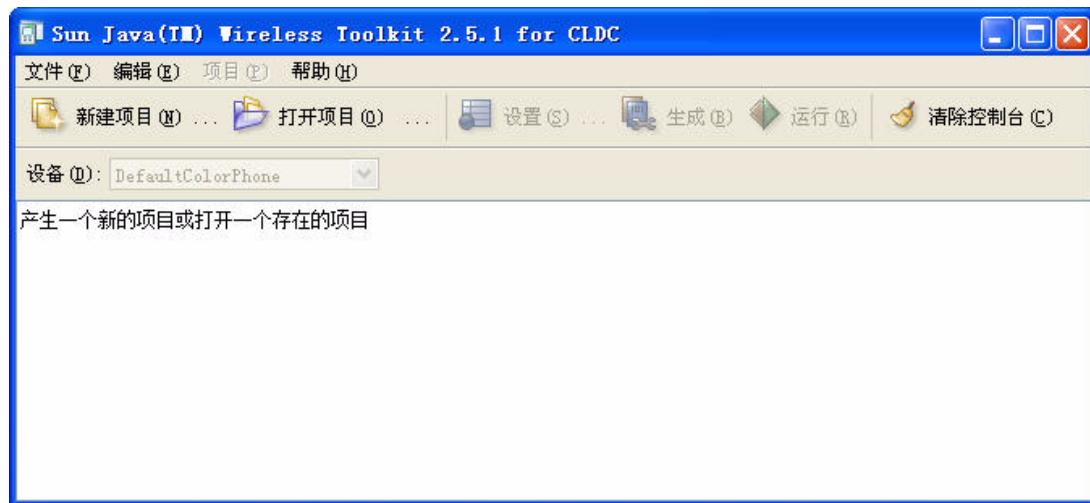
Sun Java™ Wireless Toolkit for CLDC 每次只能执行一个项目。您可以新建一个项目，也可以打开一个现有项目。

本章使用了一个非常简单的示例项目。您可以一边阅读开发周期的每个步骤，一边在工具包中逐渐深入工作。

要新建一个项目，需要先启动用户界面。在 Microsoft Windows 中，选择“开始”>“程序”>“Sun Java Wireless Toolkit 2.5.2 for CLDC”>“Wireless Toolkit 2.5.2”。¹ 将显示用户界面，如图 2-1 中所示。

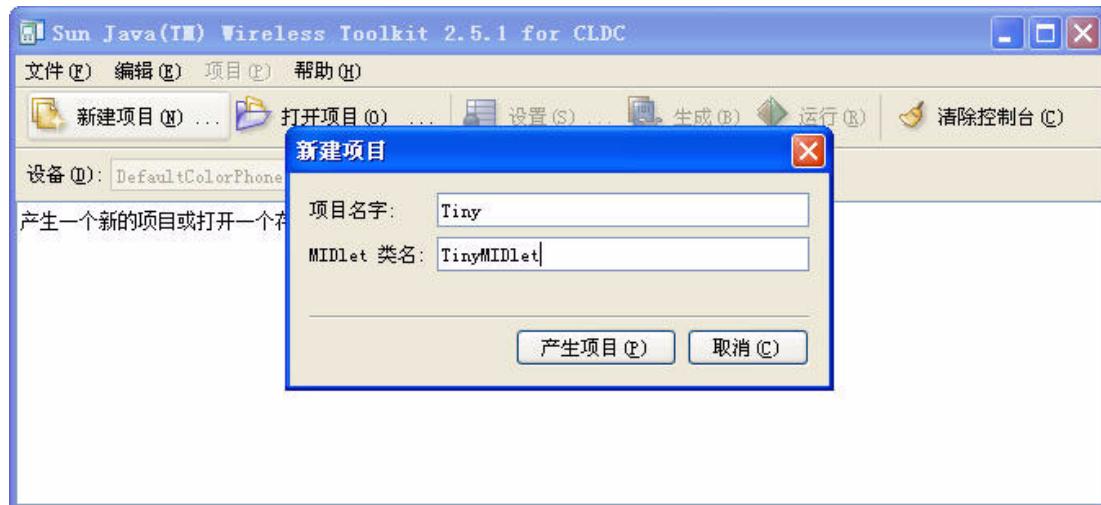
1. 根据 Microsoft Windows 的配置方式，可能需要选择“开始”>“所有程序”，而不是选择“开始”>“程序”。

图 2-1 工具包的用户界面



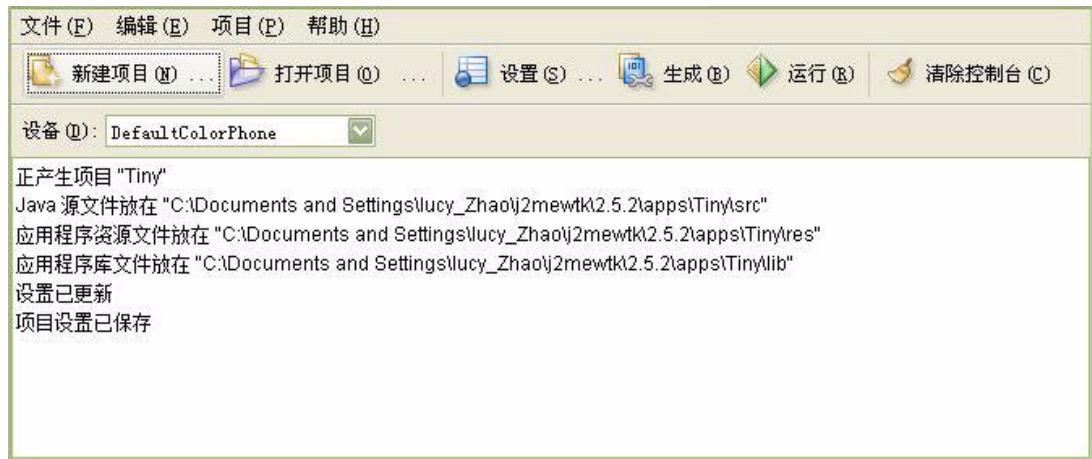
单击“新建项目”。工具包会提示您输入项目的名称和您要编写的 MIDlet 类的名称。填写名称并单击“产生项目”。

图 2-2 新建一个项目



将显示“设置”窗口。您的选择会影响项目的生成环境。默认选项适用于本示例，因此单击“确定”关闭窗口即可。在控制台中将会显示一些消息，告诉您该项目的源代码和资源文件确切的存储位置。

图 2-3 控制台中的文件位置



2.2 简单开发周期

简单开发周期类似于：

编辑源代码 > 生成 > 运行

1. 编辑源代码。

在此步骤中，创建您的应用程序将使用的 Java 源文件和资源文件。

2. 生成。

该工具包将对您的 Java 源文件进行编译和预校验。

3. 运行。

在仿真器上运行已编译的 Java 类文件。

如果在工具包尝试编译您的源文件时发生错误，请返回并再次对其进行编辑。如果您在仿真器中测试您的应用程序时发现了错误，请编辑源文件以修复该错误。

既然您进一步了解了简单开发周期，本节其余部分将介绍如何使用 Sun Java™ Wireless Toolkit for CLDC 完成每个步骤。

2.2.1

编辑源代码

编辑源代码是 Sun Java™ Wireless Toolkit for CLDC 唯一完全没有帮助的步骤。请使用所选的文本编辑器来创建和编辑源代码文件。如果没有中意的文本编辑器，请尝试使用位于 <http://jedit.org/> 的 jEdit。

如果您正按照样例项目进行操作，请新建一个 Java 技术源文件 TinyMIDlet.java。将其保存在项目的源目录中。例如，在 Windows 上，该目录为 *workdir\apps\Tiny\src\TinyMIDlet.java*。该文件包含以下简单的 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();
    }
}
```

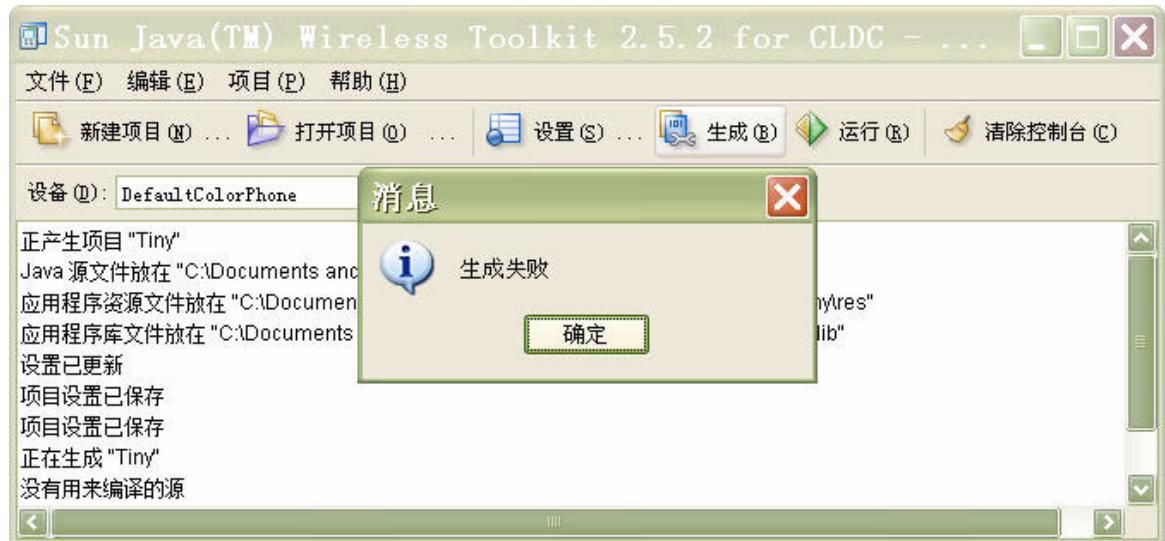
编辑完毕后保存该文件。

2.2.2 生成

接下来就是要生成源代码了。工具包使该步骤变得非常简单。

在用户界面中，单击“生成”按钮。如果您将源文件保存在正确的位置，工具包就会查找该文件并对其进行编译。将在控制台中显示编译错误。如果有错误（如图 2-4 所示），请编辑源代码以修复错误。消除错误后，控制台会通知您生成已成功完成。

图 2-4 与生成有关的消息



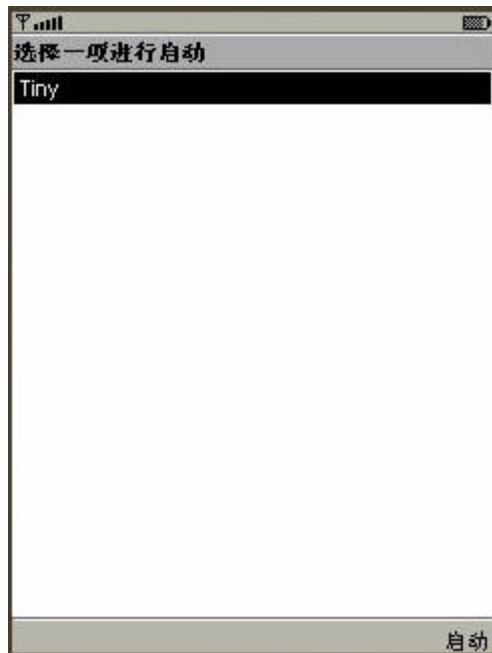
实际上，该工具包还对已编译的类文件进行预校验。在 MIDlet 类文件能在 MIDP 设备或仿真器上运行之前，必须对其进行预校验。工具包会自动为您处理此细节。有关预校验的更多信息，请参见 CLDC 规范。

2.2.3

运行

项目成功生成之后，您随时可以在仿真器中运行该项目。单击“运行”按钮。仿真器将显示项目中所有 MIDlet 的列表。

图 2-5 项目 MIDlet 的列表



选择所需的 MIDlet 并选择“启动”。如果您沿用 TinyMIDlet 示例，结果将如图 2-6 中所示。

图 2-6 TinyMIDlet 在执行中



2.3 完整开发周期

第二个开发周期稍微复杂一些。它由以下几个高级步骤组成：

编辑源代码 > 打包 > 安装 > 运行

1. 编辑源代码。

该过程与简单周期中的过程相同。

2. 打包。

在该步骤中，Sun Java™ Wireless Toolkit for CLDC 对源文件进行编译和预校验（本质上与简单周期的“生成”步骤相同）。然后，将 Java 类文件和资源文件捆绑到 MIDlet 套件 JAR 归档 (Java Archive, JAR) 文件和 MIDlet 套件描述符中。

3. 安装。

在运行 MIDlet 套件之前需要先对其进行安装。您可以将 MIDlet 套件安装到 Sun Java™ Wireless Toolkit for CLDC 仿真器或真实设备中。

4. 运行。

同简单开发周期一样，运行您的应用程序并测试是否有错误。

完整开发周期中的第一个步骤与简单开发周期相同。源代码编辑过程与往常一样。“生成”过程现在合并到了打包过程之中。

完整开发周期包括两个新步骤，即打包和安装。最后，运行已安装的应用程序与在简单开发周期中运行应用程序的方式有重要不同。

2.3.1 打包

Sun Java™ Wireless Toolkit for CLDC 能自动完成对 MIDlet 套件进行打包的任务。打包会最终产生两个文件，即一个 MIDlet 描述符和一个 MIDlet 套件 JAR 文件。描述符是一个小文本文件，包含有关 MIDlet 套件的信息。JAR 文件包含组成 MIDlet 套件的类文件和资源。在下载整个 JAR 文件之前，设备可以使用该描述符了解有关应用程序的信息，这在内存匮乏、带宽受限的无线环境中很值得重视。

要让工具包对 MIDlet 套件进行打包，请选择“项目”>“包”>“产生包”。即可生成 MIDlet 套件描述符和 JAR 文件，并将其放在项目的 bin 目录下。

打包可能涉及其他步骤。您可以使用代码混淆器来减小 MIDlet 套件 JAR 文件的大小，本章后面的内容会对该技术进行介绍。此外，该工具包还提供了若干工具，您可以利用这些工具来加密签名 MIDlet 套件。有关更多信息，请参见[第 6 章](#)。

2.3.2 安装

要正确地对 MIDlet 套件进行测试，请将该套件安装到工具包的仿真器中或真实设备中。当您在用户界面中按下“运行”按钮时，MIDlet 套件并没有安装到仿真器中，而是由仿真器直接运行 MIDlet 类。

仿真器还可以将应用程序安装到其内存中，该过程与通过无线协议技术 (Over the Air, OTA) 将应用程序传送并安装到真实设备上的过程类似。要在 Sun Java™ Wireless Toolkit for CLDC 仿真器中安装应用程序，请选择“项目”>“通过 OTA 运行”。

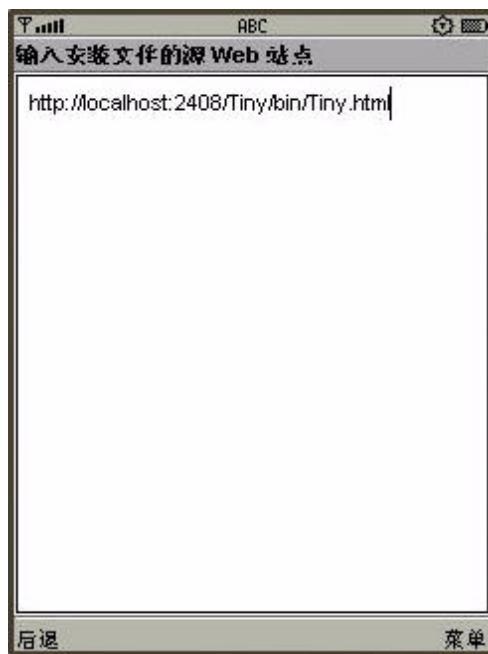
仿真器窗口将打开，但仿真器没有直接运行 MIDlet 类，而是显示应用程序管理软件 (Application Management Software, AMS) 欢迎屏幕。仿真器的软件是真实设备管理 MIDlet 套件所必须具备的软件类型的一个示例。

图 2-7 仿真器 AMS 欢迎屏幕



选择“应用程序”转到已安装应用程序的主列表。在仿真器上选择“安装应用程序”并按下“选择”按钮。仿真器提示您输入您想要安装应用程序的 URL 位置。系统已为您填写了该 URL。

图 2-8 URL 提示



从菜单中选择“执行”开始安装。仿真器显示在该 URL 下找到的应用程序的列表。仅选择其中一个，并从菜单中选择“安装”。仿真器会提示您最后一次确认您的选择。

图 2-9 确认安装



再次选择“安装”结束安装。您将返回到仿真器的已安装应用程序列表，该列表中现在已包含您刚刚安装的应用程序。

图 2-10 “应用程序”菜单



“通过 OTA 运行”是一种非常有用的机制，它使您可以轻松地在工具包仿真器上安装您的 MIDlet 套件。对包括推送注册表和签名 MIDlet 套件的安装在内的某些功能进行测试时必须使用该技术。

如果您要在真实设备上对您的 MIDlet 套件进行测试，则必须先安装该套件，这在很大程度上取决于您使用的设备。有以下两种可能性：

- 您可以将应用程序部署到 Web 服务器上，然后使用无线协议技术 (Over the Air, OTA) 将该应用程序从服务器传送到设备上，在 MIDP 2.0 规范中可以了解到该协议的相关说明。这几乎与用户在购买或安装您的应用程序时遇到的机制是相同的。
- 您可以将 MIDlet 套件传送到使用蓝牙、红外或串行连接的设备上。这比运行一台 Web 服务器要简单一些，并且虽然您无法了解到安装过程（使用 OTA 将您的应用程序安装到设备上）的细节，但是这能使您了解您的应用程序在设备上的执行情况。

2.3.3 运行

应用程序安装完成后，请从列表中选择该应用程序并从菜单中选择“启动”。

图 2-11 启动已安装的应用程序



在真实设备上运行应用程序很大程度上取决于设备本身。要了解相关信息，请查询您的设备文档。

2.4 从 MIDlet 套件创建项目

您还可以从 MIDlet 套件归档 (.jar 文件) 和描述符 (.jad 文件) 创建 Sun Java™ Wireless Toolkit for CLDC 项目。即使没有可用的源代码，这对于从工具包用户界面运行 MIDlet 套件也非常有用。您可以使用工具包轻松地操纵描述符中的属性，也可以运行项目并使用工具包的监视工具（在第 5 章中进行了介绍）以观察其行为。

要基于 MIDlet 套件创建项目，请选择“文件”>“根据 JAD/JAR 文件创建项目”。导航至要使用的描述符并单击“打开”。请注意，描述符和 JAR 文件必须位于同一目录中。

2.5

使用混淆器

混淆器是用于减小类文件大小的工具。MIDlet 套件需要压缩，这样可以尽量减少下载时间，还能满足制造商或运营商有时强行实施的对 JAR 文件大小严格限制的要求。使用混淆器是一种可使您的 MIDlet 套件 JAR 文件保持较小的方式（但不是唯一方式）。

您可以在开发周期的打包步骤中使用混淆器。虽然 Sun Java™ Wireless Toolkit for CLDC 没有附带混淆器，但它已配置为使用 ProGuard 混淆器。您只需下载 ProGuard，并将其放到工具包可以找到它的地方即可。

ProGuard 的发布遵照 GNU 通用公共许可证 (General Public License, GPL) 的条款。如果您接受该许可证的条款，您就可以免费下载并使用 ProGuard。

2.5.1

安装 ProGuard

请按照以下步骤安装 ProGuard：

1. 请转到 **ProGuard Web** 站点 <http://proguard.sourceforge.net/>。
2. 下载其最新版本。
3. 从 ProGuard 的安装目录 lib，将 proguard.jar 文件解压缩到 toolkit/bin 目录中。

2.5.2

使用 ProGuard

ProGuard 安装完毕后，可以通过选择“项目”>“包”>“产生混淆包”使用 ProGuard 了。

在某些情况下，您需要提供脚本文件来控制混淆器的工作。例如，如果您正使用 Class.forName() 装入类，那么您需要告诉 ProGuard 不要更改类名。

要调用脚本文件，您必须能够修改自己 ktools.properties 文件的副本。

- 将 toolkit/wtklib/os/ktools.properties 复制到 workdir/wtklib/os/ktools.properties。

使用文本编辑器创建一个脚本文件，然后将其保存到项目的主目录下。要了解有关脚本文件的信息，请查询 ProGuard 文档资料。下一步您需要告诉工具包如何找到该文件。要这样做，请编辑您已复制到 workdir/wtklib 中的 ktools.properties 文件。添加如下一行：

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

用脚本文件使用的名称来替换上面的 scriptfile。您必须退出并重新启动工具包才能使更改生效。

2.6 使用调试器

运行您的应用程序有另外一种方法，即利用调试器来运行它。使用调试器，您可以更密切地监视运行中的应用程序、设置断点和检查变量。

您必须自行提供调试器。您可以使用 Java SE 平台提供的 `jdb` 调试器，或者使用您选择的其他调试器。如果要使用调试器，集成开发环境 (Integrated Development Environment, IDE)（如 Sun Java Studio Mobility 软件，它并入了 Sun Java™ Wireless Toolkit for CLDC）最有可能成为您的选择。有关更多信息，请访问 <http://www.sun.com/software/products/jsmobility/>。

首先，选择“项目”>“调试”。输入调试器将用于连接仿真器的 TCP/IP 端口号，单击“调试”。仿真器开始运行，并等待来自调试器的连接。

启动调试器，将其连接到指定的端口。请确保将远程调试器设置成在远程模式下运行并使用 TCP/IP。有关更多信息，请查询调试器的文档资料。

Debugging MIDlets 包含有关将 `jdb` 用于 Sun Java™ Wireless Toolkit for CLDC 的信息。可以通过以下网址访问此文章：

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

2.7

在 Web 服务器上部署应用程序

MIDP 2.0 规范包括“无线协议用户初始化置备规范”(Over The Air User Initiated Provisioning Specification)，其中介绍了 MIDlet 套件如何通过无线协议(OTA)传送到设备。您可以使用 Sun Java™ Wireless Toolkit for CLDC 仿真器对这种情况进行测试。

要在 Web 服务器上远程部署打包的 MIDP 应用程序，请将 Java 应用程序描述符(Java Application Descriptor, JAD)文件的 MIDlet-Jar-URL 特性更改为 JAR 文件的 URL。URL 必须为绝对路径。例如：

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

接下来，请确保 Web 服务器对 JAD 和 JAR 文件使用正确的 MIME 类型：

- 对于 MIDlet 套件描述符，将 .jad 扩展名映射到 text/vnd.sun.j2me.app-descriptor MIME 类型。
- 对于 MIDlet 套件 JAR 文件，将 .jar 扩展名映射到 application/java-archive MIME 类型。

配置 Web 服务器的细节将视所用的具体软件而定。

仿真器可以实现 OTA 置备期间的设备行为。您可使用仿真器来测试和演示将 MIDlet 套件从服务器置备到设备的全过程。您所做的只是启动仿真器的 AMS。如果您已使用“通过 OTA 运行”选项，则可能您已熟悉了 AMS。

有两种方法可以启动仿真器的 AMS：

- 在 Microsoft Windows “开始”菜单中，选择“开始”>“程序”>“Sun Java Wireless Toolkit 2.5.2 for CLDC”>“OTA 置备”。
- 在命令行中，输入以下命令：

对于 Windows 系统： toolkit\bin\emulator -Xjam

对于 Linux 系统： toolkit/bin/emulator -Xjam

然后根据 AMS 提示安装应用程序。除了必须输入您自己服务器的 URL 以安装应用程序之外，其余过程与本章前面介绍的使用“通过 OTA 运行”选项的过程非常类似。

使用项目

在上一章，您学习了如何在 MIDP 开发周期借助 Sun Java™ Wireless Toolkit for CLDC 来完成任务。本章，您将进一步了解项目的详细使用方法，包括以下方面：

- 为项目选择目标 API
 - 操纵 MIDlet 套件属性，包括 MIDlet 列表
 - 了解项目的目录结构
 - 在项目中引入第三方库
-

3.1 选择 API

每个项目均是根据某个 API 集而生成的。Sun Java™ Wireless Toolkit for CLDC 支持许多 API。[第 1.6 节，第 1-7 页上的“支持的技术”](#) 中提供了详细的完整列表。通过该工具包，用户可根据要在软件上运行的设备的类型为某个 API 子集开发应用程序。

例如，虽然该工具包支持 JSR 184，即 Mobile 3D Graphics API，但用户可能希望开发不使用该 API 的应用程序。在项目的“API 选择”设置中，用户可以仅选择要使用的 API。

要了解整个过程，请启动工具包，并打开一个项目，单击“设置”以显示[图 3-1](#) 中所示的窗口：

图 3-1 项目设置窗口



在“API 选择”窗格上，“目标平台”设置用于控制该窗格其余部分的外观。选择最符合您需要的设置，并通过下面的控件调整您的选择。例如，如果要为其开发应用程序的设备符合 Java Technology for the Wireless Industry JSR，则请从组合框中选择“JTWI”。然后使用下面的控件指定 CLDC 的版本并选择可选 API。

工具包会在您编译源代码时应用您的选择。

注 - API 选择不适用于仿真器。仿真器支持所有可用 API。在项目设置中所做的 API 选择仅用于生成项目。实际上，API 选择所选的是编译和预校验源文件时工具包要使用的类路径。

3.2

更改 MIDlet 套件属性

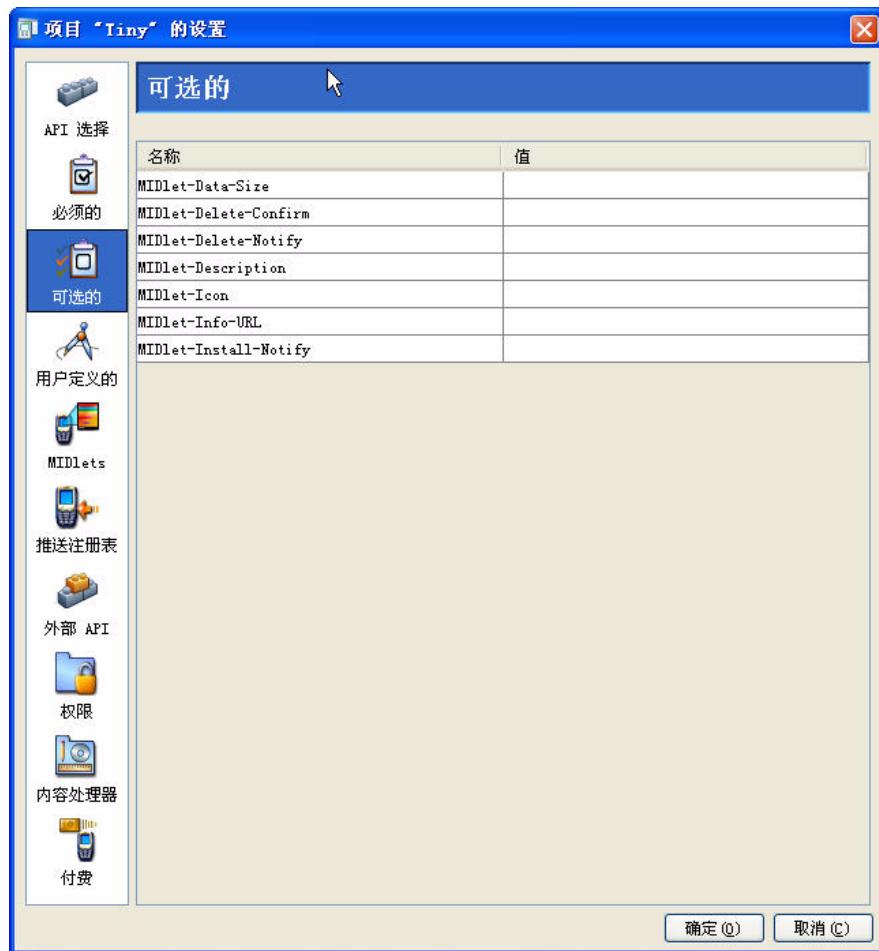
在项目设置窗口中，用户还可以控制 MIDlet 套件属性，这些属性存储在 MIDlet 套件 JAR 文件的描述符和清单文件中。

要查看这些属性，请打开项目，然后单击“设置”按钮。设置窗口左侧的图标栏中显示三个属性图标：“必须的”、“可选的”和“用户定义的”。

有关必需及可选属性的定义，请参阅 MIDP 2.0 规范最终版 2（也称为 MIDP 2.1）。Sun Java™ Wireless Toolkit for CLDC 能处理大多数细节。在开发初期，用户可能无需考虑这些属性。一旦应用程序能够稳定运行，您开始考虑在真实设备上进行部署以及将产品投放市场时，则需调整这些值。

要调整“必须的”或“可选的”窗格中的值，请单击要更改的属性关键字旁的单元。输入新值。

图 3-2 编辑 MIDlet 套件属性



要创建新的用户定义属性，请单击“用户定义的”图标。单击“添加”按钮并填写属性名称和值，然后单击“确定”。

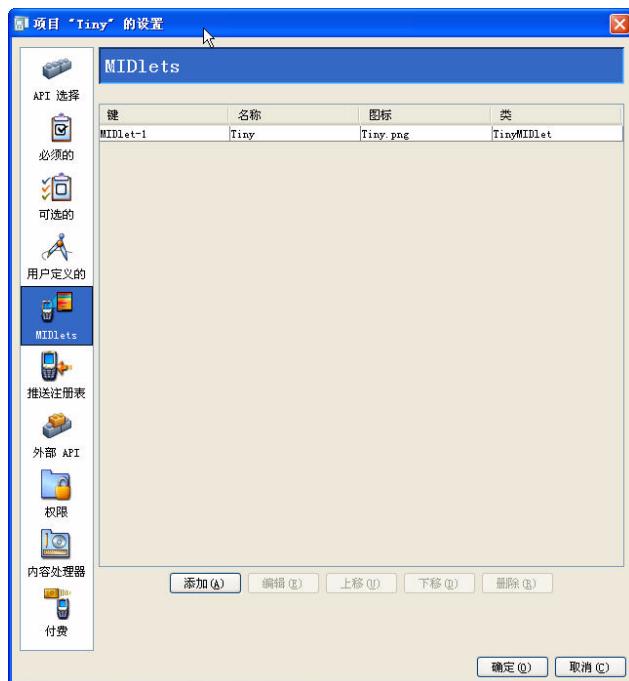
要编辑用户定义的属性值，请单击该关键字旁的值列，与编辑必需属性或可选属性的方法相同。

要删除某个属性，请选择该属性，然后单击“删除”。

3.3 操纵 MIDlet

项目设置还提供了添加或修改当前 MIDlet 套件项目中所包含的 MIDlet 的方法。要了解整个过程，请启动工具包，然后打开一个现有项目。单击“设置”，然后选择“MIDlets”图标。您将看到该项目中的所有 MIDlet 的列表。如果您刚创建了一个新项目，工具包就会自动填写第一个 MIDlet 条目。

图 3-3 项目中的 MIDlet 列表



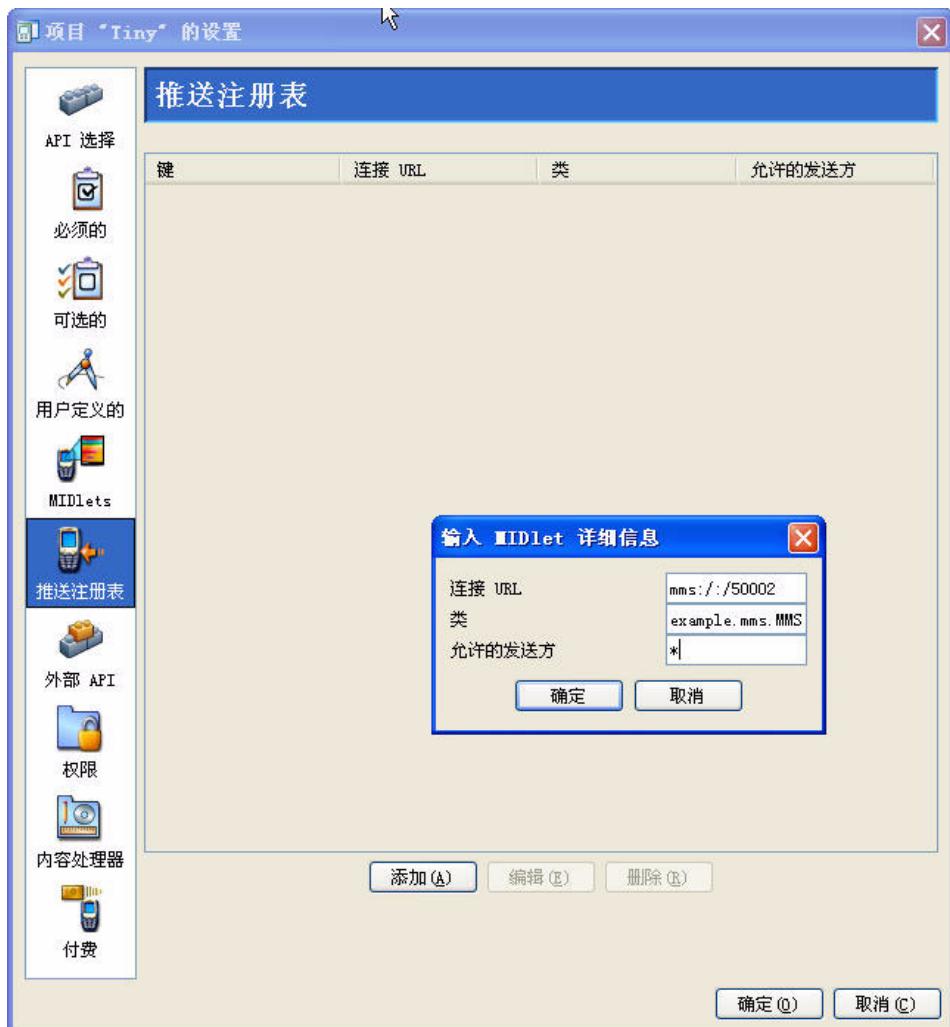
要添加新 MIDlet，请单击“添加”，然后填写名称、图标文件名和类名。如果您愿意，也可以不填写图标文件名。要更改值或删除 MIDlet 条目，请使用“编辑”和“删除”按钮。

MIDlet 名称是按 MIDlet 套件启动时显示的顺序呈现给用户的。您可以通过选择 MIDlet 并单击“上移”或“下移”来修改此顺序。

3.4 使用推送注册表

还可以通过使用项目设置来使用 MIDlet 套件的推送注册表设置。单击“设置”，然后选择“推送注册表”图标。

图 3-4 项目推送注册表设置



要向推送注册表中添加条目，请按“添加”并填写连接 URL、MIDlet 类以及允许的发送方的值，然后单击“确定”。要编辑条目，请选择该条目，然后按“编辑”按钮。要删除推送注册表条目，请选择该条目，然后单击“删除”。

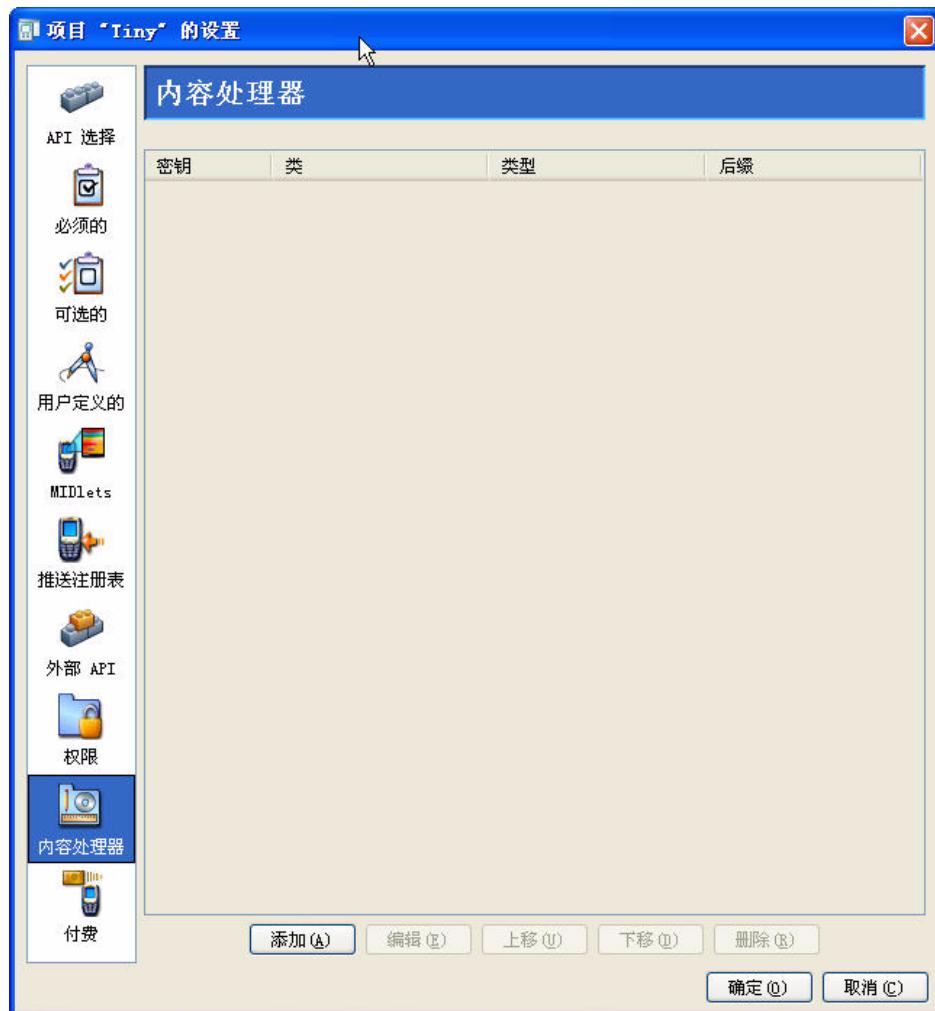
如果您确实创建了应用程序的推送注册表条目，请务必也输入相应的权限。有关详细信息，请参见[第 6 章](#)。

3.5 设置内容处理器

Sun Java™ Wireless Toolkit for CLDC 支持由 JSR 211 定义的内容处理 API (Content Handler API, CHAPI)。CHAPI 的基本概念是指 MIDlet 可依据传入内容（文件）进行启动。现代的移动电话可使用 SMS、红外、蓝牙、电子邮件及其他方法来接收内容。大多数内容都具有关联的内容类型。CHAPI 会根据特定的内容类型指定启动 MIDlet 的系统。

要修改项目中的内容处理器设置，请单击“设置”，然后选择“内容处理器”窗格。

图 3-5 配置内容处理器



列表中的每一行都代表内容处理器的设置。在本示例中，已配置了两个内容处理器，一个用于 TextViewer，一个用于 ImageViewer。要创建新的内容处理器，请按“添加”，要编辑现有内容处理器，请按“编辑”。您可以通过选择一个内容处理器并单击“上移”和“下移”按钮来调整内容处理器的顺序。要从列表中删除内容处理器，请按“删除”。

添加或删除内容处理器时，将显示详细信息窗口：

图 3-6 内容处理器详细信息



在“类”字段中，填写 MIDlet 的名称。“ID”用于调用内容处理器并控制访问的标识字符串。

“内容类型”是该内容处理器所负责的内容类型的列表。使用“添加”和“删除”管理列表。后缀字符串是 URL 后缀列表，这些 URL 后缀通常为显式内容类型的替换后缀。最后，允许的访问是内容处理器 ID 列表，这些 ID 指示了对该内容处理器具有访问权限的其他内容处理器。如果列表为空，则将向所有其他内容处理器授予该内容处理器的访问权限。

内容处理器具有相关操作，这使调用应用程序可选择处理内容的方式。例如，图像查看器内容处理器可能包括按图像的原始大小查看图像的操作以及全屏显示图像的操作。单击内容处理器详细信息窗口内的“操作”选项卡可编辑要对内容处理器执行的操作。

图 3-7 内容处理器操作



“操作”列表中包含该内容处理器的操作的内部名称。“语言环境”指显示用户可读操作名称时，可用的所有语言环境列表。“本地化的操作”是包含各个语言环境用户可读操作名称的表格。每一行表示一个语言环境，并且操作按列列出。用户可通过读取一行来查看某一特定语言环境所有用户可读操作的名称。

3.6 项目目录结构

项目有一个标准的目录结构。项目本身由 `apps` 子目录中的某个目录表示。例如，在 Windows 上，`demos` 项目位于 `workdir\apps\demos` 中。下表对项目目录结构进行了说明。

表 3-1 项目目录结构

| 目录 | 描述 |
|-------------------------|--|
| <code>bin</code> | 打包项目时，会将 MIDlet 套件描述符和 JAR 文件置于该目录下。该目录还包含未被打包的清单信息，可能还包含一个 HTML 文件，供使用“通过 OTA 运行”时内部使用。 |
| <code>classes</code> | 该目录用于存储已编译的类文件。 |
| <code>lib</code> | 该目录中用于存储第三方库以在该项目中包括该库。 |
| <code>res</code> | 该目录用于存储图像、声音以及其他资源文件。它们被打包放入 MIDlet 套件 JAR 文件的根目录下。 |
| <code>src</code> | 该目录用于存储源文件。 |
| <code>tmpclasses</code> | 供工具包使用。 |
| <code>tmplib</code> | 供工具包使用。 |

此外，项目目录还包含 `project.properties` 文件，其中包含有关该项目的信息。要从项目中删除临时目录和文件，请选择“项目”>“清除”。

3.7 使用第三方库

Sun Java™ Wireless Toolkit for CLDC 允许用户在应用程序中并入第三方库。第三方库可提供用户不想自己生成的功能，从而减少开发时间，但用户需时刻注意 MIDlet 套件 JAR 文件的大小。

在应用程序中使用第三方库时，JAR 会根据第三方库的大小而变大。用户可使用混淆器来减小代码大小，良好的混淆器甚至还可以除去不使用的库部分。即使使用混淆器，第三方库也可能比您自己从头开始精心编写的自定义代码大。您必须在减少开发时间和 MIDlet 套件 JAR 文件的大小之间权衡利弊。

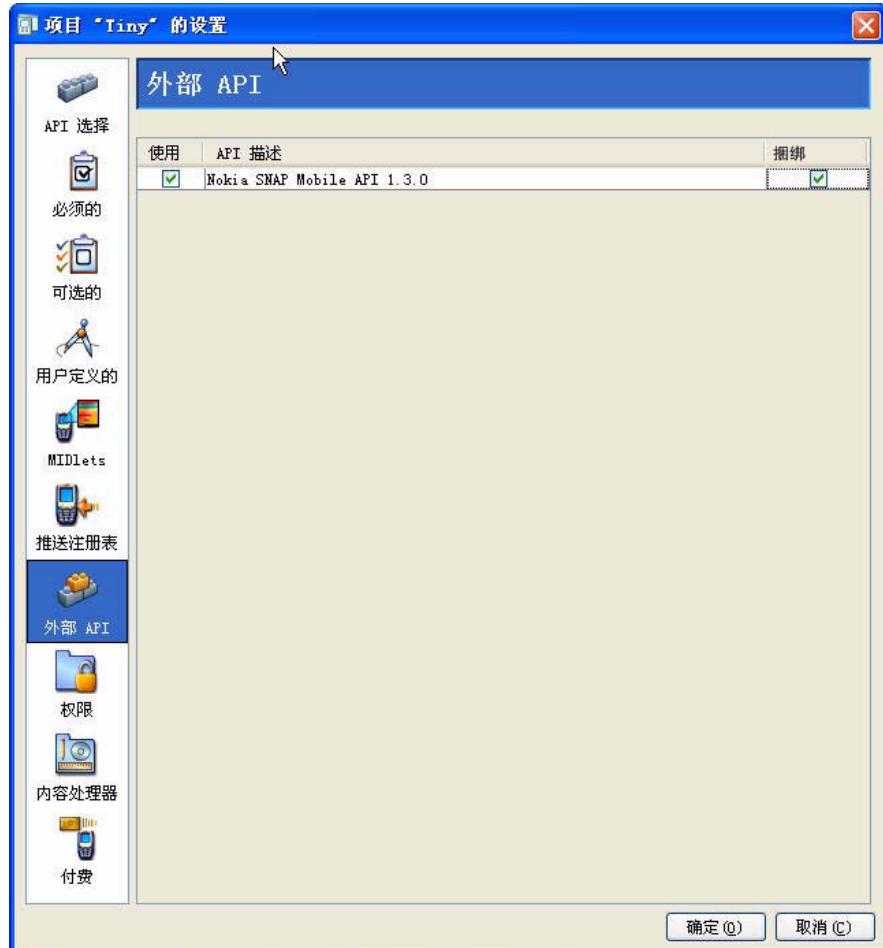
Sun Java™ Wireless Toolkit for CLDC 提供了两种并入第三方库的方法。项目设置中的“外部 API”窗格使用户可轻松在项目中并入库或排除库。此外，您还可以在特定位置放置库，使一个或所有项目得以使用。

3.7.1 使用外部 API

要指定应包括的 API，请单击“设置”，然后单击“外部 API”图标。将显示可用外部 API 的列表（该列表可能为空）。

要在生成时将 API 添加到类路径中，请选中“使用”列中的框（请参见图 3-8）。如果还希望将 API 捆绑到应用程序中，还需选中“捆绑”框。如果要部署应用程序的设备上不存在所选的外部 API，则需在应用程序中捆绑这些 API。

图 3-8 选择外部 API



3.7.2 用于一个项目的第三方库

生成和打包项目时，将使用项目的 lib 目录下所存储的库文件。库文件应为 Java 技术类的 JAR 或 Zip 文件。

因此，如果在 Windows 上执行安装，并且应用程序的名称为 Tiny，则类库位于 *workdir\apps\Tiny\lib* 中。生成、运行、调试和打包项目时，将使用 lib 目录下的类文件。

3.7.3 用于所有项目的第三方库

某些设备的库可供所有已安装的 MIDlet 套件使用。例如，某制造商可以使其所有设备均支持附加的 API。在这种情况下，用户需要在生成和测试应用程序时可以使用这些库。由于安装 MIDlet 套件的设备本身就具有这些库，因此用户不希望将这些库包含在打包的 MIDlet 套件中。

只需将这些库放入 *workdir\apps\lib* 目录下即可。所有项目均可使用该目录下的库。

3.8 配置 Wireless Toolkit

工具包包括某些高级配置选项。可通过编辑位于以下位置的 *ktools.properties* 文件的副本来使用这些选项：

对于 Windows 系统: toolkit\wtklib\Windows\ktools.properties

对于 Linux 系统: toolkit/wtklib/Linux/ktools.properties

将 *ktools.properties* 复制到 *workdir/os/wtklib*，然后按照本章其余部分所述的内容进行更改。

- [更改控制台字体](#)
- [设置应用程序目录](#)
- [设置 javac 编码属性](#)
- [使用修订控制系统](#)

下次启动工具包时，可以看到对 *ktools.properties* 所进行更改的效果。

3.8.1 更改控制台字体

可通过编辑两个属性来更改控制台（以及其他文本区域）中所使用的字体。此处为将字体更改为 Times New Roman 20 号的示例：

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

要恢复默认字体和大小，请删除这两个属性。

3.8.2 设置应用程序目录

默认情况下，Sun Java™ Wireless Toolkit for CLDC 会将应用程序存储在工作目录的 `apps` 子目录下。在 `ktools.properties` 中以如下形式添加一行，即可更改这一默认设置：

`kvem.apps.dir: 应用程序目录`

对于 Windows，目录路径中的所有反斜杠 (\) 字符前都必须再加一个反斜杠。另外，目录路径不能包含任何空格。

例如，要将应用程序目录设置为 `D:\dev\midlets`，请使用：

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

Linux 路径可以像往常一样指定。

3.8.3 设置 `javac` 编码属性

默认情况下，`javac` 使用的是当前运行的 Java SE 平台中的编码集。有关如何覆盖默认源文件编码的信息，请参见[附录 C](#)。

3.8.4 使用修订控制系统

使用 `filterRevisionControl` 属性，可将工具包配置为识别和忽略由 SCCS、RCS 和 CVS 修订控制系统创建的辅助文件。

要识别和忽略辅助文件，应该在 `ktools.properties` 中包含下面一行：

```
kvem.filterRevisionControl: true
```

这样可以防止工具包将修正控制文件视为源和资源文件。例如，工具包将名为 `src\\SCCS\\s.MyClass.java` 的文件视为 SCCS 修订控制文件，而不是 Java 技术源文件。

使用仿真器

Sun Java™ Wireless Toolkit for CLDC 仿真器用于在您的台式机上模拟 MIDP 设备。这样您就可以方便地看到您的应用程序在 MIDP 环境中的执行情况，同时也为您提供了一个紧密的开发周期，在您的台式机上即可完成整个开发周期。

仿真器并不代表某个特定的设备，而是提供对其所支持的 API 的正确实现。

4.1 仿真器样机

样机是仿真器实现之上的一个薄层，为仿真器提供特定的外观、屏幕特征和输入控件。Sun Java™ Wireless Toolkit for CLDC 附带表示不同设备类型的样机。

表 4-1 仿真器样机

| 名称 | 屏幕大小 | 画布大小 | 颜色 | 输入 |
|-------------------|-----------|-----------|------|--------|
| 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 |

如果您愿意，您可以创建自己的仿真器样机。有关详细信息，请参见基本自定义指南。

4.2 仿真器控制

仿真器的外观和功能就像标准桌面窗口中的一部移动电话。本节介绍如何控制仿真器。描述和图示均以 DefaultColorPhone 样机为例，但所有样机操作方法与其类似。

图 4-1 DefaultColorPhone 仿真器样机



可以使用鼠标单击这些按钮以将其按下。大多数按钮还具有便于使用的键盘快捷键。键盘数字键 0 到 9 与仿真器的 0 到 9 按钮相对应。一些不明显的键盘快捷键如下表中所示。

表 4-2 键盘快捷键

| 仿真器按钮 | 键盘键 |
|-------|-------|
| 左软按钮 | F1 |
| 右软按钮 | F2 |
| 电源按钮 | Esc |
| 选择按钮 | Enter |

文本输入的操作与在许多真实设备上的操作非常相似。多次按下某个数字键，即可获得所需的字母。例如，按下 5 键两次得到字母 K。当您输入文本时，星号键 (*) 用于在大写、小写、数字和符号之间进行切换。屏幕顶端的指示器显示您当前所处的模式。井号键 (#) 用于输入空格。

另外，您也可以只按下键盘上的键来输入文本。虽然这样输入文本很方便，但您必须记住您的用户很可能无法享受这种便利条件。

另一种便利的功能是在文本区域复制和粘贴信息。同时按下 Ctrl+V，可将剪贴板中的文本粘贴到 TextBox 或 TextField。要复制 TextBox 或 TextField 中的内容，请同时按下 Ctrl+C。此操作可将文本字段的完整内容放置在剪贴板。

4.3 设置仿真器首选项

您可以对仿真器的设置进行调整，使其更接近特定设备，或在不同的资源情况下测试您的应用程序。

4.3.1 网络代理

仿真器使用您的台式机网络连接。例如，如果仿真器运行建立 HTTP 连接的 MIDlet，则该仿真器将尝试使用台式机网络设置来建立 HTTP 连接。

如果您的开发计算机受防火墙保护，您可以使用代理服务器建立 HTTP 连接。如果您不能确定，请试着检查您的浏览器设置，查看它是否使用了代理服务器。

如果您正在使用代理服务器，那么您就需要配置仿真器以使用相同的代理服务器。要执行此操作，请选择“编辑”>“首选项”。在“网络配置”窗格上，填写您要使用的代理服务器的名称和端口号。还可以选择希望使用的 HTTP 版本。

如果代理服务器使用 HTTP 基本验证（请参见 RFC 2617），请检查验证并填写用户名和密码。

4.3.2 存储大小

您可以设置或修改指定给每个仿真器的永久存储容量，也可以更改堆栈大小（应用程序的对象所允许的存储容量）。选择“编辑”>“首选项”，然后选择“存储”项。

图 4-2 存储首选项



4.3.2.1 永久存储

仿真器具有永久存储，默认情况下，位于电话样机目录中的 appdb 子目录下。这些文件的扩展名为 .db。

例如，在 Windows 上，仿真器样机 DefaultColorPhone 的永久存储存储在 `workdir\appdb\DefaultColorPhone\manager_storage_settings.db` 中。

如果同时运行同一个仿真器样机的多个实例，工具包将为每个实例生成唯一的文件路径。例如，在 Windows 上，DefaultColorPhone 的实例可能具有文件路径名 `workdir\appdb\temp.DefaultColorPhone1`、`workdir\appdb\temp.DefaultColorPhone2` 等等。

注 - 文件 `workdir\appdb\DefaultColorPhone\in.use` 会记录标记为使用中的存储根目录的数目。如果仿真器出现故障，您需要删除 `in.use` 文件。

工具包使您能够为存储文件选择一个不同的位置，并能限制该存储的大小。如果您希望在少量永久存储可用的情况下测试应用程序的性能，可以使用该特性。

要调整永久存储设置，请选择“编辑”>“首选项”，并在左窗格中单击“存储”。在“存储根目录”字段中，输入要用于永久存储的目录的名称。您可以仅输入相对路径，此时您指定的目录将创建在 `appdb` 子目录中。

默认情况下，您可以使用大小为 1 MB (1024 KB) 的永久存储。您可以输入以 KB 为单位的大小限制。请记住，除了您应用程序使用的空间之外，存储实现还有一些系统开销。例如，如果您输入的永久存储大小为 8KB，则您的应用程序数据和存储系统开销可用大小为 8192 字节。

要删除仿真器的永久存储，请选择“文件”>“实用程序”。然后单击“清除数据库”按钮，清除永久存储。清除数据库不影响已安装的应用程序。

4.3.2.2 堆栈大小

堆栈是用于存储您的应用程序对象的内存。要更改堆栈的大小，请选择“编辑”>“首选项”，然后选择“存储”项（请参见图 4-2）。默认情况下，堆栈大小为 1 MB。可以对最大堆栈大小进行设置，使其更接近模拟真实设备的情况。在“堆栈大小”字段中填写最大堆栈大小（以 KB 为单位）。

4.3.3 调整仿真器性能

仿真器使用台式机的许多资源，包括其显示器和网络连接。与基于台式机的仿真器相比，真实 MIDP 设备的处理器速度通常较低、内存较小、网络连接较慢，而且显示器的类型也可能不同。

Sun Java™ Wireless Toolkit for CLDC 允许您对真实设备的受限环境进行模拟。虽然仿真器不代表真实设备，但是调整性能设置也为您提供了有用的信息，使您了解到您的应用程序在各种运行时环境中的执行情况。

选择“编辑”>“首选项”，然后在左窗格中单击“性能”。请参见图 4-3。

图 4-3 调整仿真器的性能



调整“图形的基本时延”可以改变应用程序调用 `Graphics` 类中的绘图方法至开始绘图之间所经过的时间。

要更改屏幕特征，请选择一种“显示刷新”类型。如果选择“定时”类型，那么您还需要指定“刷新率”。

要模拟真实设备较低的速度，请选中“启用虚拟机速度仿真”并选择希望的速度。

要调整模拟的网络速度，请选中“启用网络吞吐量仿真”，然后选择速度（以位/秒为单位）。

4.4 暂停和恢复

MIDlet 有一个由 MIDP 规范定义的生命周期。MIDlet 可由设备启动或停止。此外，外部事件（如传入电话呼叫）也会导致设备暂停某个 MIDlet。

仿真器提供了暂停和恢复运行中 MIDlet 的简单机制。这对在应用程序暂停时测试其性能非常有用。

当仿真器运行时，从仿真器窗口菜单中选择 "MIDlet" > “暂停”。运行中的 MIDlet 将暂停，屏幕将显示“正在传入呼叫 ...”消息。

要恢复操作，请从菜单中选择 "MIDlet" > “恢复”。

4.5 单独运行仿真器

开发期间，您会经常通过按下“运行”按钮或选择“项目”>“通过 OTA 运行”直接从工具包运行仿真器。为了进行测试或演示，您可能希望单独运行仿真器。本节将介绍一些不同的方法。Sun Java™ Wireless Toolkit for CLDC 安装程序创建的程序组包括用于单独运行仿真器的几个选项。

- 要直接运行应用程序，请选择“运行 MIDP 应用程序 ...”项，这和按下“运行”按钮效果一样。工具包提示您在本地磁盘上查找 MIDlet 描述符文件。请注意，必须还得有相应的 MIDlet 套件 JAR。
- 要运行仿真器的应用程序管理软件 (Application Management Software, AMS)，请选择“OTA 置备”项，这和用户界面中的“通过 OTA 运行”功能基本相同。将弹出仿真器，同时出现 AMS 欢迎屏幕，通过键入 URL 您就可以安装应用程序了。
- 要更改仿真器的首选项，请从工具包程序组中选择“首选项”项。与从用户界面中选择“编辑”>“首选项 ...”效果一样，都弹出同样的首选项窗口。
- 不运行用户界面也可以访问 Sun Java™ Wireless Toolkit for CLDC 实用程序，只需选择“实用程序”项。
- 最后，您可以更改默认情况下使用的仿真器样机。选择“默认设备选择”项，并选择其中一个可用的仿真器样机。下次启动仿真器时，即使用所选的样机。

您还可以通过命令提示符来运行仿真器。有关更多信息，请参见[附录 B](#)。

4.6

使用第三方仿真器

第三方公司，如设备制造商和无线运营商，有时会创建与工具包兼容的设备仿真器。通过将其他仿真器安装到工具包中，您可以获得在更多实现上运行应用程序的经验。具体过程通常是，首先解压缩或安装第三方仿真器，然后将其目录复制到 `workdir\wtklib\devices` 中。下次运行工具包时，就可以使用该仿真器了。

以下网址列出了目前可用的部分仿真器：

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

监视应用程序

Sun Java™ Wireless Toolkit for CLDC 提供了若干用于监视应用程序性能的工具。这些工具能够帮助您调试和优化代码：

- 事件探查器列出了应用程序中每个方法的使用频率和执行时间。
- 内存监视器显示应用程序运行时内存的使用情况。
- 网络监视器显示应用程序传送和接收的网络数据。它支持多种网络协议，包括 HTTP、HTTPS、SMS 和 CBS。
- 跟踪将底层信息输出到工具包控制台。

注 – 监视功能可能会降低应用程序的执行速度。

5.1 使用事件探查器

事件探查器记录应用程序中的每个方法。对某次特定的应用程序运行，事件探查器都计算出每个方法占用的时间，以及调用每个方法的次数。当应用程序结束运行并关闭仿真器后，将弹出事件探查器，您可以从中浏览所有的方法调用信息。

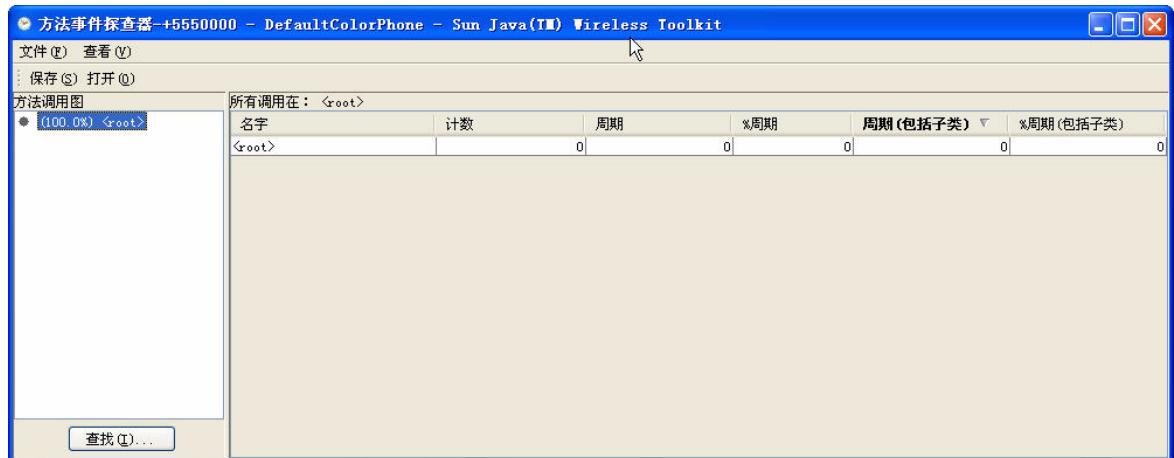
要打开事件探查器，请选择“编辑”>“首选项”。请参见图 5-1。在左窗格中单击“监视”。在右窗格中选中“启动 Profiling”。如果您想查看所有系统实现方法的配置处理信息，请选择“显示系统类”。否则，事件探查器将只显示包含对应用程序方法调用的系统方法。单击“确定”。

图 5-1 打开事件探查器



现在单击“运行”按钮启动应用程序。以通常的方式与您的应用程序进行交互操作。结束后，关闭仿真器。事件探查器弹出，并显示有关应用程序中所有方法调用的信息。

图 5-2 方法事件探查器



事件探查器显示两种类型的信息：

- 方法关系显示在名为“方法调用图”的层次结构列表中。
- 事件探查器的右侧显示每个方法及其子代的执行时间和调用次数。

注 – 从仿真器中获得的配置处理值反映不出真实设备上的实际值。

5.1.1 方法调用图

方法调用图显示方法调用的层次结构。调用其他方法的方法显示为文件夹。双击某个方法将其打开，并查看该方法所调用的方法。没有调用任何其他方法的方法显示为灰色圆圈。

您可以搜索某个特定类或方法名。单击“查找”，并填写名称。搜索将从方法调用图中的当前选择开始执行，直到最后。如果您想搜索完整的方法调用图，请在单击“查找”按钮之前选中“环绕”。

在方法调用图中单击其他节点时，事件探查器的右侧就会显示该节点方法的详细信息。

5.1.2 执行时间和调用次数

事件探查器窗口的右侧显示有关方法的详细信息。您可以看到方法名、方法调用次数以及仿真器使用该方法所用的时间。执行时间有四种不同的描述方法：

- 周期表示方法本身所用的处理器时间。
- % 周期表示方法本身所用总执行时间的百分比。
- 周期（包括子类）表示某方法及其调用的方法所用的时间。
- % 周期（包括子类）表示某方法及其调用的方法所用的时间占总执行时间的百分比。

单击任意一列，以按照该列进行排序。再次单击该列，可以在升序和降序之间切换。

右窗格显示方法调用图中当前所选节点中包含的方法。如果您想查看每个方法，请在方法调用图中单击 `<root>` 节点。

5.1.3 保存和装入事件探查器信息

要保存您的事件探查器会话，请在事件探查器窗口中单击“保存”按钮。然后选择一个文件名。

要装入事件探查器会话，请选择“文件”>“实用程序”。单击“事件探查器”，然后按“启动”。选择文件后，事件探查器窗口会出现，并显示所有会话信息。

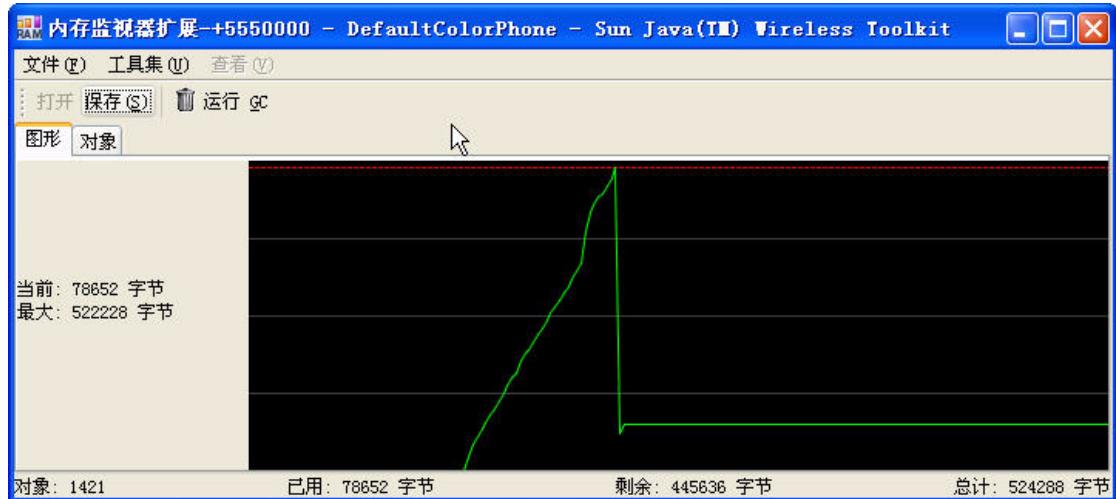
5.2 使用内存监视器

许多 MIDP 设备上的内存都不够用。Sun Java™ Wireless Toolkit for CLDC 中的内存监视器使您可以轻松地检查应用程序内存的使用情况。您可以查看应用程序使用的内存总量，以及每个对象的内存使用情况详细列表。

要打开内存监视器，请选择“编辑”>“首选项”。然后在左窗格中单击“监视”。选择“启用内存监视器”。

下次运行仿真器时，就会显示内存监视器窗口，同时显示一幅随时间变化的应用程序内存使用情况图。由于创建的每个对象都被记录下来，因此内存监视器会使应用程序的启动速度变慢。

图 5-3 内存监视器图



内存监视器图显示以下信息：

- **当前**——应用程序当前使用的内存量。
- **最大**——自程序开始执行起使用的最大内存量，在图形中用中断的红线表示。
- **对象**——堆栈中的对象数。
- **已用**——已用的内存量。
- **剩余**——尚未使用的可用内存量。
- **总计**——启动时可用的内存总量。

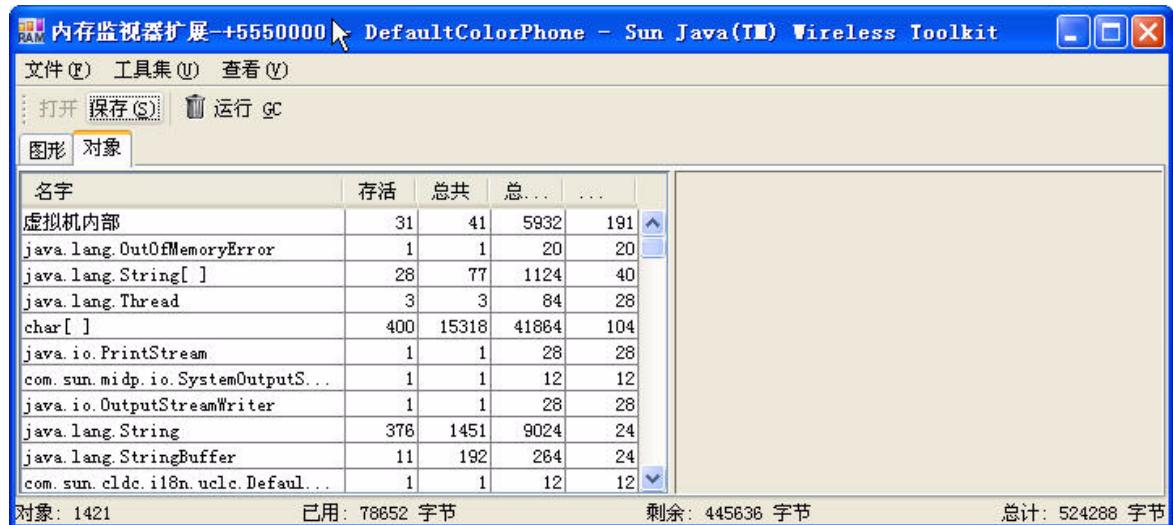
请记住，要修改堆栈大小，可以选择“编辑”>“首选项”，然后选择“存储”选项卡。有关详细信息，请参见第 3 章。

要让系统执行垃圾回收，请单击“运行 GC”。

注 – 在仿真器上看到的内存使用情况与真实设备上看到的不完全相同。请记住，仿真器并不能代表真实设备。它仅仅是其所支持的 API 的一种可能实现。

要查看有关应用程序中对象的详细信息，请在内存监视器窗口中单击“对象”选项卡。

图 5-4 内存监视器对象显示



将显示一个包括以下列的表：

- **名字**——对象类名称。
- **存活**——实例数目。其中有些可以进行垃圾回收。
- **总共**——自应用程序开始运行以来所分配的对象总数。
- **总计大小**——对象使用的内存总量。
- **平均大小**——对象的平均大小，由总计大小除以存活实例数目得到。

单击任意一列标题，按照该列进行排序。

通过从内存监视器窗口菜单中选择“查看”>“查找...”，可以搜索某个特定的类名。

5.2.1 保存和装入内存监视器信息

要保存内存监视器会话，请单击“保存”按钮，然后选择一个文件名。

要装入内存监视器会话，请选择“文件”>“实用程序”。单击“内存监视器”，然后按“启动”。选择文件后，内存监视器窗口会出现，并显示所有会话信息。

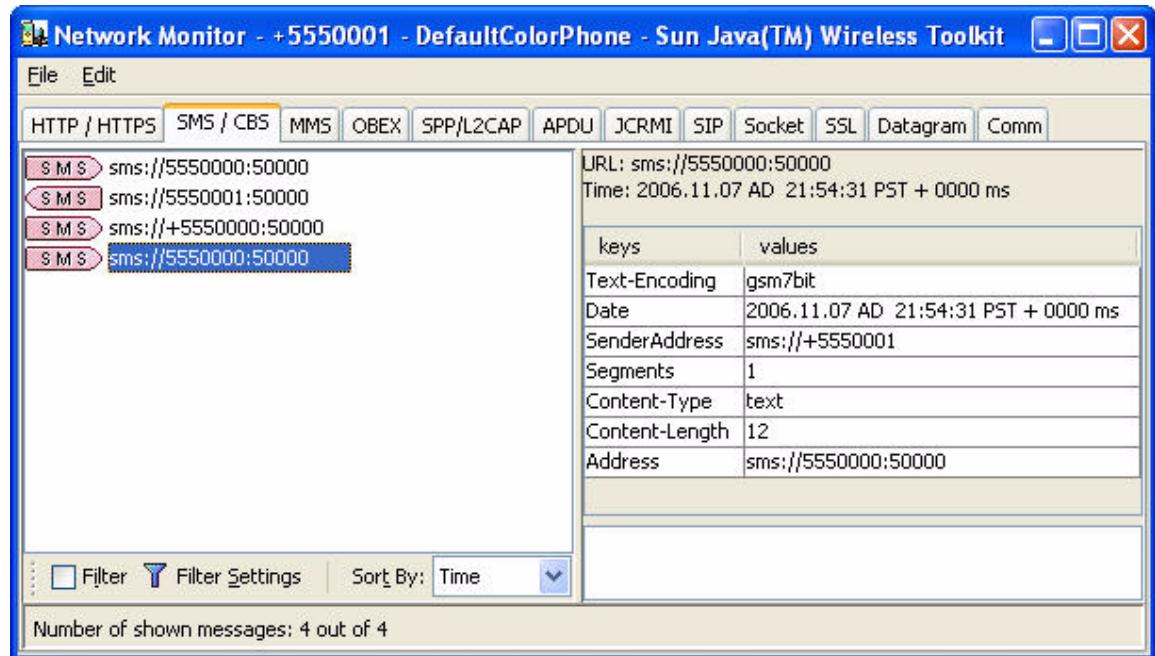
5.3 使用网络监视器

MIDP 应用程序至少能进行 HTTP 网络连接，同时也可进行其他类型的网络连接。网络监视器提供了一种查看应用程序通过网络发送和接收信息的便捷方式。此方法可以帮助您对网络交互进行调试或寻找优化网络通信流量的途径。

要打开内存监视器，请选择“编辑”>“首选项”。在左窗格中单击“监视”。选中“启用网络监视”。

下次您运行仿真器时，就会显示网络监视器窗口。

图 5-5 网络监视器



应用程序进行任何类型的网络连接时，将捕获并显示有关该连接的信息。图中显示了 HTTP 请求和响应。

屏幕左侧显示消息和消息片段的层次结构。单击某条消息或消息的某一部分，可在网络监视器的右侧查看其详细信息。双击某条消息或消息的各部分，可展开或收拢它们。

消息主体以原始十六进制值及对应文本进行显示。

注 - 您可对尚处于发送过程的消息进行检查。未完成发送的消息在消息树中以粗体高亮显示。

5.3.1 过滤消息

过滤器用于检查网络通信总流量的某个子集。过滤设置针对于当前使用的网络协议。

要使用过滤器，请按下 "Filter Settings" 按钮。根据您的需要，更改过滤设置。

表 5-1 网络监视器过滤设置

| 网络协议 | 过滤设置 |
|------------------|---|
| HTTP/HTTPS | 输入要与 HTTP 消息的各部分进行匹配的文本，这些部分包括：URL、状态行、标头或主体。例如，在 URL 字段中输入 slashdot 将仅显示其 URL 包含 slashdot 的消息。 |
| SMS/CBS | 可以指定要匹配的协议、消息类型和方向。此外，还可以输入要与发送方、接收方以及消息内容进行匹配的文本。 |
| MMS | 输入要与方向、发送方、接收方以及抄送和密件抄送接收方进行匹配的文本。此外，还可以按照主题、内容 ID、内容位置、MIME 类型和编码进行过滤。 |
| OBEX、SPP/L2CAP | 您可以使用 URL 或标头内容进行过滤。 |
| APDU、JCRMI | 对 URL 或消息内容进行过滤。 |
| SIP | 无可用 |
| 套接字、SSL、数据报、Comm | 输入要与连接字符串 (URL) 或内容进行匹配的文本。 |

完成过滤设置输入后，按下“确定”，返回到网络监视器。“Filter”复选框处于选中状态，表明正在使用某个过滤器。要禁用过滤器并查看所有消息，请取消选中该复选框。

5.3.2 对消息进行排序

要使消息树按某种特定顺序排列，请单击 "Sort By" 组合框并选择一种标准。

- **Time**——消息将按照发送或接收时间顺序进行排序。
- **URL**——消息将按 URL 地址进行排序。同一地址的多条消息将按时间进行排序。
- **连接**——消息将按照通讯连接进行排序。使用同一连接的消息将按时间进行排序。通过此排序类型可以查看按请求和相关响应进行分组的消息。
- 排序参数取决于您选择的消息协议。例如，按时间排序与使用套接字协议的消息无关。

5.3.3 保存和装入网络监视器信息

要保存网络监视器会话，请从网络监视器窗口菜单中选择“文件”>“保存”，或者“文件”>“另存为”，然后选择一个文件名。

要装入网络监视器会话，请选择“文件”>“实用程序”。从列表中选择“网络监视器”，然后按“启动”。选择文件后，网络监视器窗口会出现，并显示所有会话信息。

5.3.4 清除消息树

要从网络监视器中删除所有消息，请从网络监视器菜单中选择“编辑”>“清除”。

安全和 MIDlet 签名

MIDP 2.0 (JSR 118) 包含一套基于保护域的全面安全模型。MIDlet 套件安装在保护域中，保护域可决定受保护功能的访问权。MIDP 2.0 规范还包括推荐的使用公钥密码学检验和验证 MIDlet 套件的方法。

有关权威性信息，请参阅 MIDP 2.0 规范。有关使用 Sun Java™ Wireless Toolkit for CLDC 进行 MIDlet 签名的概述，请参阅文章 Understanding MIDP 2.0's Security Architecture，位于以下网址：

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

如果要了解有关公钥密码学的更多背景知识，请参阅文章 MIDP Application Security 1: Design Concerns and Cryptography，位于以下网址：
<http://developers.sun.com/techtopics/mobility/midp/articles/security1/>

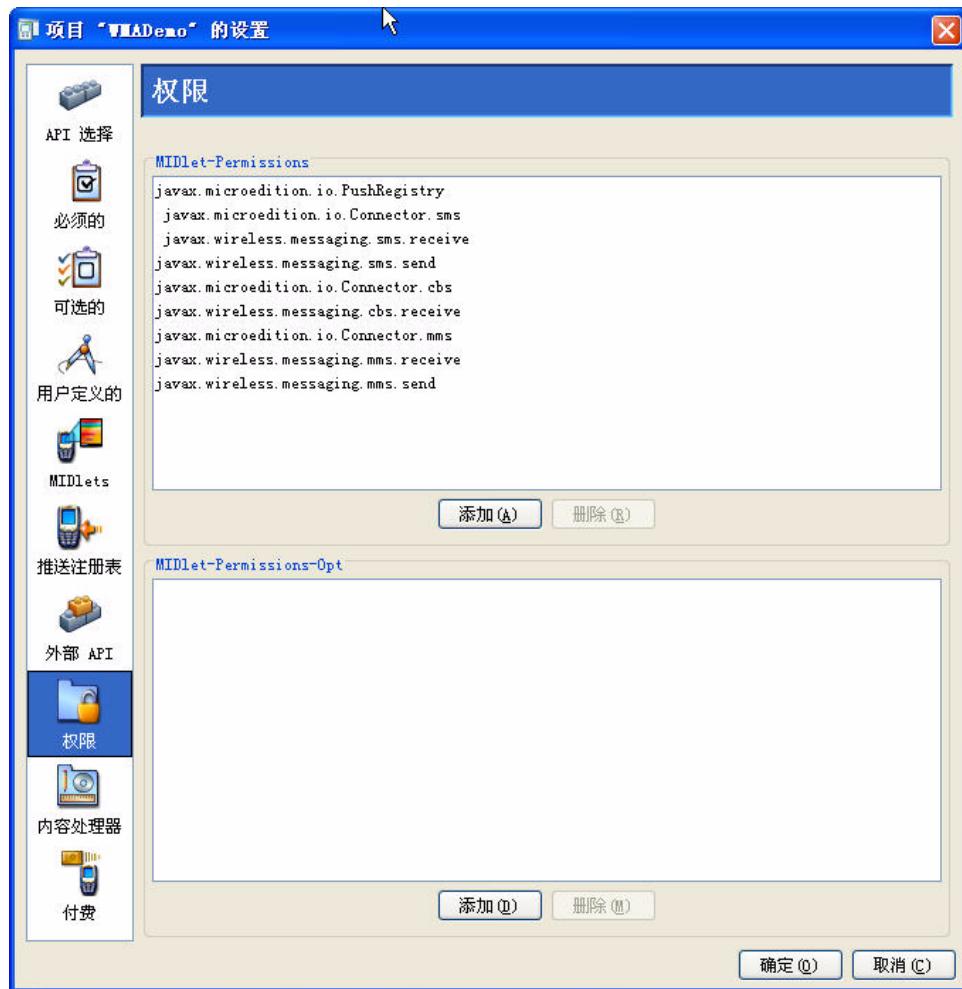
本章介绍 Sun Java™ Wireless Toolkit for CLDC 对保护域、权限以及 MIDlet 签名的支持。

6.1 权限

MIDlet 必须具有执行敏感操作的权限，例如连接到网络。权限具有特定的名称，MIDlet 套件可通过 MIDlet 套件描述符中的属性来指示其所需的某些权限。

用户可通过单击“设置”按钮向项目中添加这些权限属性。选择“权限”图标。“MIDlet-Permissions”框中显示 MIDlet 必须具有的权限，“MIDlet-Permissions-Opt”框中包含的是可选权限。

图 6-1 MIDlet 套件权限



要向任意一个框中添加权限，请单击“添加”，然后选择要添加的权限。要删除某项权限，请高亮显示该权限，然后单击“删除”。

6.2 选择安全策略

Sun Java™ Wireless Toolkit for CLDC 支持通过 JSR 185 (Java Technology for the Wireless Industry) 和 JSR 248 (Mobile Service Architecture, 即 MSA) 定义的安全策略。第 6.2.1 节, 第 6-3 页上的“MSA 保护域”和第 6.2.2 节, 第 6-4 页上的“Java for the Wireless Toolkit Industry 保护域”中进一步介绍了保护域。

要选择希望仿真器使用的安全策略, 请选择“编辑”>“首选项”, 然后在“类别”列表中选择“安全”。从“安全策略”组合框中, 选择“MSA”或“JTWI”。选择其中一个可用安全策略。

在使用“通过 OTA 运行”时, 会将打包的 MIDlet 套件直接安装到仿真器中, 并在安装时将其置于保护域中。仿真器使用公钥密码学来确定已安装 MIDlet 套件的保护域。

如果 MIDlet 套件未签名, 则会将其置于默认的保护域中。MSA 和 JTWI 的默认保护域不同。请参见第 6.2.1 节和第 6.2.2 节。如果 MIDlet 已签名, 则会将其置于与签名密钥证书链的根证书相关联的保护域中。

例如, 假设有家名为 Respectable Software 的公司, 要分发加密签名的 MIDlet 套件。Respectable Software 则从假设的证书授权机构 Super-Trustee 购买签名密钥对。Respectable Software 使用签名密钥签名 MIDlet 套件, 并将其证书连同 MIDlet 套件一起分发。当 MIDlet 套件安装到仿真器或某个设备上时, 实现将使用其自己的 Super-Trustee 根证书副本验证 Respectable 的证书。然后, 其会使用 Respectable 证书验证 MIDlet 套件上的签名。假设一切检查无误, 设备或仿真器就会将 MIDlet 套件安装到与 Super-Trustee 的根证书相关联的保护域中, 多数情况为 `identified_third_party`。

该工具包提供用于签名 MIDlet 套件、管理密钥和管理根证书的工具。

6.2.1 MSA 保护域

该工具包支持 MSA 的五个保护域:

- `unidentified_third_party`——为无法确定来源和真实性的应用程序提供高级别的安全措施。当应用程序尝试执行敏感操作时, 用户会不断地收到提示。
- `identified_third_party`——用于已使用加密证书确定其来源的 MIDlet。将不会自动授予权限, 但用户收到提示的频率将小于 `unidentified_third_party` 域。
- `manufacturer`——用于其凭据源自制造商的根证书的 MIDlet 套件。
- `minimum`——在该域中, MIDlet 没有任何权限。
- `maximum`——在该域中, 可向 MIDlet 授予所有权限。

按“运行”按钮运行仿真器中的应用程序时, 默认情况下, 代码将在 `unidentified_third_party` 保护域中运行。

6.2.2 Java for the Wireless Toolkit Industry 保护域

Sun Java Wireless Toolkit 包含四个保护域：

- **untrusted**——为无法确定来源和真实性的应用程序提供高级别的安全措施。当应用程序尝试执行敏感操作时，用户会不断地收到提示。
- **trusted**——在该域中，可向 MIDlet 授予所有权限。
- **minimum**——在该域中，MIDlet 没有任何权限。
- **maximum**——在该域中，可向 MIDlet 授予所有权限（相当于 **trusted**）。

按“运行”按钮运行仿真器中的应用程序时，默认情况下，代码将在 **untrusted** 保护域中运行。

6.3 签署 MIDlet 套件

要签名 MIDlet 套件，必须首先将其打包（选择“项目”>“打包”），然后选择“项目”>“签名”。屏幕将显示签名窗口（图 6-2）。

图 6-2 “签署 MIDlet 套件”窗口



在“别名列表”中选择要使用的密钥，然后单击“签署 MIDlet 套件”按钮（绿色三角形）即可。

6.4 管理密钥

MIDlet 签名窗口还可用于管理密钥。出于测试目的，用户可创建密钥对以签名 MIDlet。要在设备上进行部署，必须从该设备所认可的证书授权机构获得签名密钥对。您也可以从现有的 Java SE 平台密钥库中导入密钥。

6.4.1 创建新密钥对

要创建全新的密钥对，请单击“新建密钥对”。工具包会提示您输入密钥别名，以及将与该密钥对关联的信息。

图 6-3 创建新密钥对



单击“创建”后，工具包会提示您选择保护域。刚创建的密钥对和保护域之间似乎没有直接关系，但这样非常有意义：

- 工具包会使用刚创建的密钥对来创建一个自签名的根证书。
- 将此根证书会被添加到仿真器的根证书列表中。
- 工具包需要将此根证书与保护域建立关联。

现在，设想一下安装使用新密钥签名的 MIDlet 套件后的情况：

- 实现检查 MIDlet 套件描述符中的证书链。在这种情况下，该证书链只有一个证书，即自签名根证书。
- 实现尝试在其内部列表中查找证书链的根证书。由于已在创建密钥对时添加了根证书，因此会找到根证书。
- 实现在确定该证书有效后会使用它来验证 MIDlet 套件上的签名。
- MIDlet 套件会安装到用户指定的任何保护域中。

6.4.2 获得真实密钥

在 Sun Java™ Wireless Toolkit for CLDC 环境中创建密钥对并签名 MIDlet 的功能仅供测试使用。在真实设备上运行应用程序时，必须从该设备所认可的证书授权机构获得签名密钥对。

使用真实密钥签名 MIDlet 套件的步骤如下：

1. 生成新的密钥对。

在 Sun Java™ Wireless Toolkit for CLDC 中，用户可通过在 MIDlet 签名窗口中按“新建密钥对”来生成新的密钥对，如上所述。

2. 生成证书签名请求 (Certificate Signing Request, CSR)。

a. 在签名窗口中按“生成 CSR”。

b. 要更改 CSR 文件的位置，请输入新路径，或按“浏览”，然后选择新的文件位置。

c. 按“创建”编写 CSR 文件。

编写 CSR 文件后，屏幕将显示表明创建成功的消息。

3. 将 CSR 发送给证书授权机构 (Certificate Authority, CA)。

CA 需要用户提供更多信息以便验证您的身份。用户还必须向 CA 支付其为您生成证书的费用。

CA 验证您的身份并收取费用后，其会发送用于证实您的公钥的证书。

4. 通过在 MIDlet 签名窗口中按“导入证书 ...”，将该证书导入到 Sun Java™ Wireless Toolkit for CLDC 中。

现在，用户即可使用其自己的私钥来签名 MIDlet 套件了。Sun Java™ Wireless Toolkit for CLDC 将处理将签名和证书置于 MIDlet 套件中时所涉及到的细节问题。

6.4.3 导入现有密钥对

在 Java SE 平台密钥库中，可能具有用户要用于 MIDlet 签名的密钥。在这种情况下，用户需要将签名密钥导入到 Sun Java™ Wireless Toolkit for CLDC 中，以便签名 MIDlet 套件。要执行此操作，请从 MIDlet 签名窗口中单击“导入密钥对”。选择包含 Java SE 平台密钥库的文件。系统将提示用户选择要导入的密钥对的别名，然后用户只需在此密钥对导入密钥库后，输入密钥对的别名以便识别。最后，必须为该密钥对的根证书选择保护域。

6.4.4 删 除密钥对

要从 MIDlet 签名窗口删除密钥对，请选择其别名，然后选择“操作”>“删除选择”。

6.5 管理证书

本节介绍如何使用 Sun Java™ Wireless Toolkit for CLDC 管理仿真器的根证书列表。

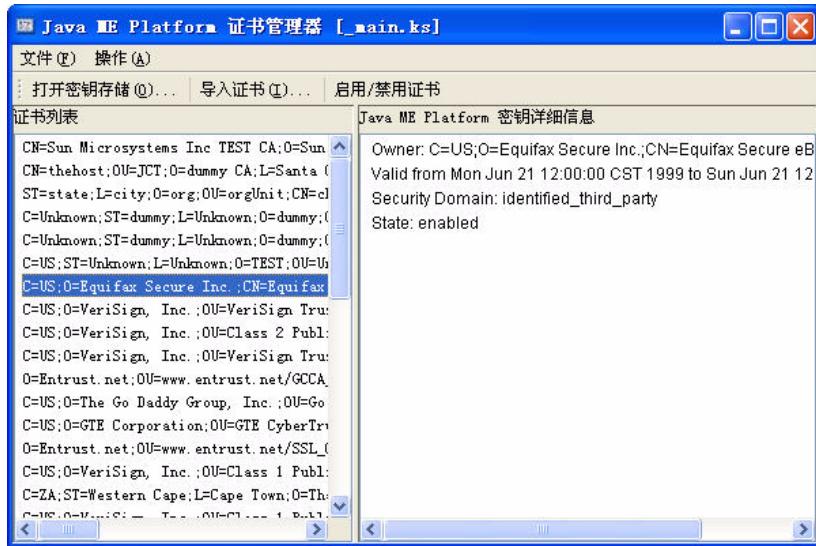
真实设备具有类似的根证书列表，尽管用户通常不能修改这些列表。如果要在真实设备上部署应用程序，必须使用设备上已存在其根证书的证书授权机构颁发的签名密钥。否则，设备将无法验证您的应用程序。

在开发应用程序时，工具包的证书管理实用程序提供了一种操纵仿真器根证书列表的便捷方法，以满足测试需求。

选择“文件”>“实用程序”。选择“管理证书”，然后按“启动”打开“证书管理器”窗口。将显示微型密钥库 `_main.mks`。该密钥库位于 `appdb` 目录中。

`appdb` 目录还包含 `keystore.ks` 和 `serverkeystore.ks`。Java ME 平台证书管理器无法打开 `*.ks` 文件，但是您可以按照第 6.5.2 节，第 6-8 页上的“导入证书”中所述从这些密钥库中导入证书。

图 6-4 证书管理器



在窗口左侧的“证书列表”中，每个证书均显示为单独的一行。单击某个证书时，窗口右侧将显示其详细信息以及该证书的关联保护域。

6.5.1 启用和禁用证书

证书可以启用也可以禁用。如果要使证书暂时不可用而不将其从密钥库中删除，这是很方便的功能。要启用或禁用证书，请在列表中选择该证书，然后按“启用 / 禁用证书”。工具包将提示您确认该操作。选择“是”继续。

6.5.2 导入证书

用户可导入证书文件或从 Java SE 平台密钥库文件中的证书。

要导入文件中的证书，请在“证书管理器”窗口中单击“导入证书”。找到要导入的证书文件后，选择与该证书相关联的保护域。

要从 Java SE 平台密钥库中导入证书，请在“证书管理器”窗口的菜单中选择“操作” > “导入 Java SE 证书”。首先，为证书选择保护域。然后选择密钥库文件，再输入密钥库口令。最后，选择要导入的证书的别名。

6.5.3 删 除证书

要从列表中删除证书，请选择该证书，然后选择“操作” > “删除选择”。

6.6 USB 令牌支持

USB 令牌为公钥和私钥以及证书提供了可移植且受密码保护的存储。Java SE PKCS#11 本机界面支持访问具有与 PKCS#11 兼容的本机驱动程序的 USB 令牌。安装驱动程序后，将包含 PKCS#11 库。在 Windows 上，库为 win32 DLL。

本节提供了在 Windows 平台上安装和使用 USB 令牌的样例说明。

注 - Linux 不受支持，因为尚未使用 Linux 驱动程序对 USB 令牌进行全面测试。如果与 PKCS#11 兼容的本机驱动程序可用，则 USB 令牌可以在 Linux 上工作。

本节其余部分逐步介绍了 USB 令牌支持所需的安装和设置过程。

6.6.1 安装 USB 令牌驱动程序

关闭所有应用程序。

1. 转至 <http://downloads.geotrust.com/TCSPIKEY0407203016.exe>
2. 当“文件下载”对话框打开时，单击“保存”。
请记下保存可执行文件的目录。
3. 选择可执行文件，然后双击以启动 **Crypto Token** 安装。
按照安装提示进行操作。插入令牌以便完成安装。
4. 将启动“Windows 新硬件向导”。
按照说明进行操作，并接受所有默认操作。
5. 向导结束后，选择“是”以重新启动计算机。

重新设置 USB 令牌口令

该步骤仅对新 USB 令牌有效。所有 USB 令牌均具有默认的口令 PASSWORD（全部大写）。您应重设该口令。

1. 单击 Windows 任务栏上的“开始”。选择“所有程序”>“GeoTrust 令牌”>“iKey 2000 系列软件”>“口令实用程序”。
2. 单击“更新口令”。
在输入新口令之前，系统将提示您输入“旧口令”。
3. 重设口令。
GeoTrust 和 Cingular 建议将口令设置为至少由八个混合字符组成。

管理 USB 令牌

转至 GeoTrust 驱动程序安装目录。运行 `CIPUtils.exe` 以管理 USB 令牌的内容。

6.6.2 使用 USB 令牌

“签署 MIDlet 套件”对话框提供了对 USB 令牌的访问。连接了 USB 令牌并安装驱动程序之后，选择“文件”>“装入密钥库”>“从 USB 令牌”（或键入 Ctrl -T）。如果 USB 令牌受密码保护，则系统将提示您输入密码。正确装入令牌后，将列出所有别名和密钥详细信息。然后即可选择密钥，并按照[第 6.3 节，第 6-4 页上的“签署 MIDlet 套件”](#)中所述将其用于签名。

当尝试从 USB 令牌装入密钥库时，可能会看到错误消息。

- 如果无法访问 USB 令牌，则可能会看到错误“可能未插入 USB 令牌或未安装驱动程序，或驱动程序无效”。请确保已插入令牌。您可能需要尝试使用计算机上的其他 USB 端口。
- 如果未找到本机库，系统将提示您输入与驱动程序一起安装的 DLL 库的路径。其通常位于驱动程序的安装目录。

使用 Wireless Messaging API

Sun Java™ Wireless Toolkit for CLDC 利用成熟的模拟环境来支持 Wireless Messaging API (WMA)。WMA 1.1 (JSR 120) 使 MIDlet 可以发送和接收短消息服务 (Short Message Service, SMS) 或小区广播服务 (Cell Broadcast Service, CBS) 消息。WMA 2.0 (JSR 205) 也支持 MMS 消息。

本章介绍可用于开发 WMA 应用程序的工具。首先介绍如何配置仿真器的 WMA 支持。接下来介绍 WMA 控制台，一种用于测试 WMA 应用程序的便利实用程序。本章末尾简要介绍了网络监视器的 WMA 支持。

7.1 设置仿真器电话号码

每个运行中的仿真器实例都有一个模拟电话号码，它显示在仿真器窗口的标题栏中。这些电话号码很重要，因为它们被用作 WMA 消息地址。默认情况下，第一个仿真器实例的电话号码为 +555000，后续仿真器实例的号码均是唯一的并按升序排列：+5550001、+5550002、+5550003 等。

您可以通过选择“编辑”>“首选项”，然后在左窗格中选择“WMA”来更改指定的电话号码。

图 7-1 设置 WMA 首选项



“下一个仿真器的电话号码”字段取其字面含义。如果在此字段中填写一个号码，则下一个仿真器实例将使用该号码。

如果已使用“下一个仿真器的电话号码”，或该字段为空，则下一个仿真器实例将使用“首次分配的电话号码”。后续实例依此类推。

例如，假设您在“下一个仿真器的电话号码”中输入 +6269333，而在“首次分配的电话号码”中输入 +5550000，如果启动四个仿真器实例，则其电话号码将依次为 +6269333、+5550000、+5550001 和 +5550002。

7.2 模拟不可靠的网络

长消息被分割成片段，然后再分别发送这些片段，全部接收完毕后再将这些片段重装为长消息。您可以在 Sun Java™ Wireless Toolkit for CLDC 中模拟无线网络的某些不可靠方面。与前面一样，请选择“编辑”>“首选项”，然后选择“WMA”。

如果希望工具包丢失某些消息片段，请将“随机消息片段丢失”滑块调整为所需的百分比。要模拟发送和接收消息片段过程中的延迟，请在“消息片段传递延迟”字段中输入以毫秒为单位的延迟时间。

7.3 通过 WMA 控制台发送消息

WMA 控制台是一种便利的实用程序，用户可以使用它发送和接收消息。例如，用户可使用 WMA 控制台向仿真器上运行的某个 MIDlet 发送 SMS 消息。

要启动 WMA 控制台，请选择“文件”>“实用程序”。单击“WMA 控制台”，然后按“启动”。

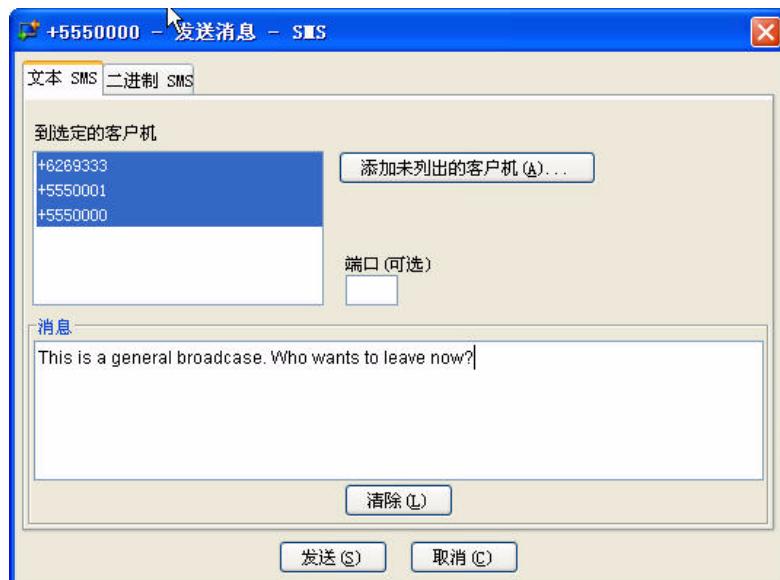
图 7-2 WMA 控制台



7.3.1 发送文本 SMS 消息

要发送文本 SMS 消息，请单击“发送 SMS”，屏幕将显示发送窗口。

图 7-3 发送文本消息



该窗口自动列出所有正在运行的仿真器实例的电话号码。选择目的地（按住 Ctrl 键，同时单击鼠标选择多个目的地），并输入端口号（如果需要）。键入消息，然后单击“发送”。

7.3.2 发送二进制 SMS 消息

您可以使用 WMA 控制台将文件内容作为二进制消息发送。单击“发送 SMS”显示发送窗口。单击“二进制 SMS”选项卡。

图 7-4 发送二进制消息



接收方选择方式与发送文本 SMS 消息时相同。用户可直接键入文件的路径，或单击“浏览”打开文件选择器。

7.3.3 发送文本或二进制 CBS 消息

发送 CBS 消息与发送 SMS 消息类似，只是不用选择收件人。要发送文本或二进制 CBS 消息，请在 WMA 控制台中单击“发送 CBS”。屏幕将显示“发送”窗口。

图 7-5 发送 CBS 消息



7.3.4 发送 MMS 消息

MMS 消息通常包含一个或多个图像或声音文件。MMS 消息可发送给多个收件人。要从 WMA 控制台中发送 MMS 消息，请单击“发送 MMS”按钮。

用于编写 MMS 消息的窗口有两个选项卡，一个针对收件人，一个针对内容。请先填写主题和收件人。如果要添加更多收件人，请单击“添加”按钮。例如，要将消息发送给正在运行的号码为 +5550001 的仿真器，请在“到”这一行中填写 mms://+5550001。要删除某个收件人，请先选择其所在的行，然后单击“删除”。

图 7-6 添加 MMS 消息的收件人



要在消息中添加媒体文件，请单击“部分”选项卡。单击“添加”将某个部分添加到消息中。要删除某个部分，请选择该部分，然后按“删除”。

图 7-7 向 MMS 消息中添加部分



7.4

在 WMA 控制台中接收消息

WMA 控制台也可接收消息。WMA 控制台窗口在标题栏中具有其自己的电话号码。您可以将消息从正在仿真器上运行的应用程序发送到 WMA 控制台。

接收的消息显示在 WMA 控制台的文本区域中。

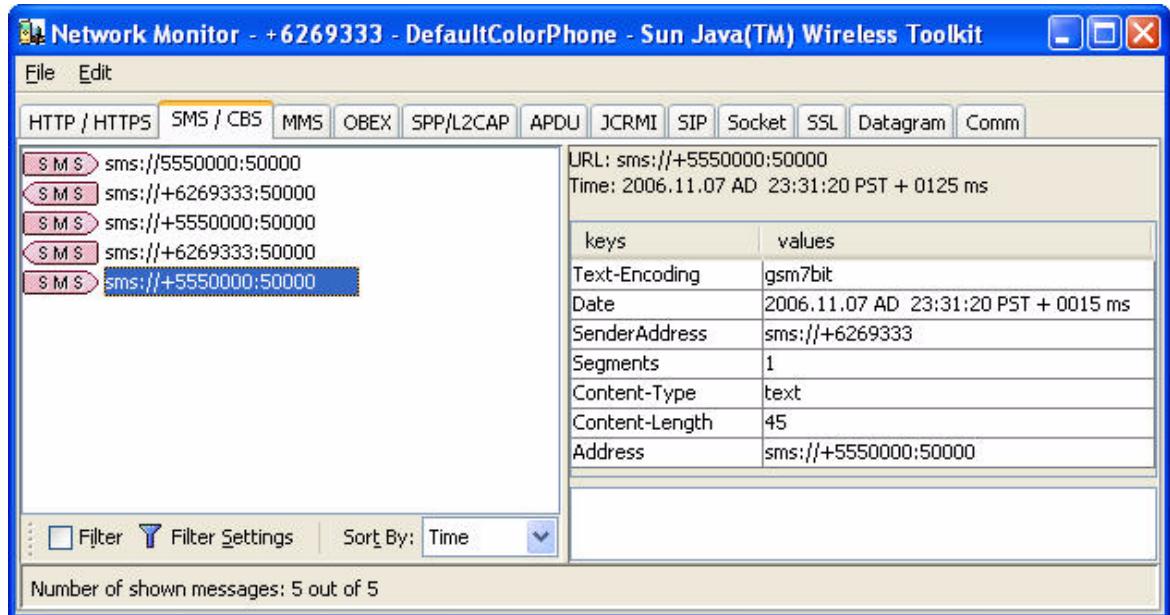
7.5

结合使用网络监视器与 WMA

[第 5 章](#) 中对网络监视器进行了全面介绍。您可以使用网络监视器跟踪从仿真器发送或由仿真器接收的 WMA 消息。

单击 "SMS/CBS" 或 "MMS" 选项卡查看 WMA 消息。有关这些消息及其片段的信息显示在网络监视器的左窗格中。单击某一消息或消息片段可在右窗格中查看其详细信息。

图 7-8 使用网络监视器查看 WMA 消息



使用 Mobile Media API

Mobile Media API (MMAPI) (JSR 135) 提供了一个标准 API，用于呈现和捕获基于时间的媒体，如音频和视频。该 API 在各种设备支持的媒体格式、协议和功能等方面设计得非常灵活。有关使用 MMAPI 编程的信息，请参见以下文章：

- Mobile Media API Overview

http://developers.sun.com/techtopics/mobile/apis/articles/mmapi_overview/

- The J2ME Mobile Media API

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

8.1 支持的格式和协议

仿真器的 MMAPI 实现支持下列媒体类型。

表 8-1 支持的 MMAPI 媒体格式

| MIME 类型 | 描述 |
|----------------------|------------------|
| audio/amr | 自适应多速率 |
| audio/midi | MIDI 文件 |
| audio/sp-midi | 可扩展多和弦 MIDI |
| audio/x-tone-seq | MIDP 2.0 单音序列 |
| audio/x-wav | WAV PCM 采样音频 |
| image/gif | GIF 89a (动画 GIF) |
| video/mpeg | MPEG 视频 |
| video/vnd.sun.rgb565 | 视频捕获 |

8.2 自适应多速率 (Adaptive Multi-Rate, AMR) 内容

Sun Java™ Wireless Toolkit for CLDC 模拟对自适应多速率 (Adaptive Multi-Rate, AMR) 内容 (<http://www.ietf.org/rfc/rfc3267.txt>) 的支持。虽然工具包无法对 AMR 内容进行解码，但实现可以在请求时返回用于 AMR 内容的播放器。

8.2.1 Windows

在 Windows 上，AMR 文件将被转换为常规 WAVE 文件，然后传递到 Qsound。因为 Windows 版本界面具有 3GPP 实现功能，因此您不必进行任何操作即可播放 AMR 文件。

8.2.2 Linux

Linux 的 AMR 支持基于 3GPP AMR-NB (AMR Narrow Band) 引用实现解码器和 SOX 音频处理器（不支持 AMR-WB (AMR Warrow Band)）。

8.2.2.1 启用 AMR 支持

请按照以下步骤启用 AMR 支持。

1. 获取由 3GPP 提供的 AMR-NB RI。

有若干可用的版本。其中一个版本可在以下位置找到：

http://www.3gpp.org/ftp/Specs/archive/26_series/26.073/26073-530.zip

2. 打开包中包含的 `makefile`。

找到以 "CFLAGS = " 开头的行。

a. 添加 `DMMS_IO` 选项。

b. 删除 `pedantic-errors` 选项。

保存并编译。如果遇到问题，请尝试删除 O3 (或 O2) 优化标志 (二进制数据大小将接近二倍)。

3. 要生成 RI，请输入： `make VAD=VAD1`

编译之后，您将得到名为 `decoder` 的二进制文件。即为 AMR-NB 解码器本身。

4. 设置环境变量 AMR_DECODER 以指向解码器的路径。

例如，如果 ~/amr 是到解码器的路径，请指定：

```
export AMR_DECODER=~/amr/decoder
```

5. 设置 decoder 文件的执行访问权限。

在与解码器相同的目录下，键入：

```
chmod 555 ./decoder
```

6. 按如下方式对解码器进行测试：

运行 MobileMediaAPI 演示 Simple Player，如第 A.13 节，第 A-30 页上的“MobileMediaAPI”所述。从主菜单中，选择 "Simple Player"，然后选择 "AMR Narrow Band [jar]"。请勿忘记打开扬声器。

8.2.2.2 AMR 格式支持

您也可能遇到一些 AMR-NB 的“克隆”问题。AMR-NB 编解码器数据似乎至少有两种文件格式：

- .AMR 文件，以“AMR 文件存储格式”存储

该格式在 draft-ietf-avt-rtp-amr-10.txt 第 6.2 节中详细说明，涵盖在 Ericsson AMR 转换器工具中。（草稿为 RFC 3267 的早期形式）。这些文件由 Ericsson AMR 工具和 Nokia Series 60 电话进行处理。它们包含标头 #!AMR\n，并按大端字节存储顺序进行编码。

- .COD 文件，以“AMR 接口格式 2”存储

该格式在附录 A 的 "3GPP TS 26.101" 中详细说明。这些文件由 3GPP TS 26.104 浮点引用编解码器源包进行编码和解码。它们不包含标头并按小端字节存储顺序进行编码。

要在这些格式之间进行转换，请尝试以下 Python 脚本：

```
http://www.connactivity.com/~eaw/amrwork/amrconv.py
```

8.3 使用 MediaControlSkin

Sun Java™ Wireless Toolkit for CLDC 附带了一个仿真器样机 MediaControlSkin，该样机用于多媒体的回放和控制。该样机包括一些按钮，按钮上带有表示播放、停止、放大音量、减弱音量以及其他命令的符号。要了解 MediaControlSkin 的功能，请使用 MobileMediaAPI 演示应用程序测试该样机。

8.4 媒体捕获

Sun Java™ Wireless Toolkit for CLDC 仿真器支持音频和视频捕获。通过使用仿真器运行时所在的系统的捕获功能来支持音频捕获。

通过模拟照相机输入来支持视频捕获。

有关演示如何捕获音频和视频的源代码及详细信息，请参见 [MobileMediaAPI](#) 示例应用程序。

8.5 正常的 MIDlet

MIDlet 有一个在 MIDP 规范中定义的生命周期。MIDlet 会因传入电话呼叫之类的事件而暂停。正常的 MIDlet 会在暂停后释放重要的设备资源并在 MIDlet 恢复后重新分配或重新启动这些资源。在 MMAPI 处理区域，在 MIDlet 暂停时停止正在呈现内容的所有 Player。

如果暂停 MIDlet 的同时没有停止其正在运行的 Player，则 Sun Java™ Wireless Toolkit for CLDC 将在控制台上显示一条消息。您可以使用 [MobileMediaAPI](#) 演示应用程序中的暂停音频测试 MIDlet 来测试此功能。有关详细信息，请参见 [附录 A](#)。

该警告消息对于每个运行中的仿真器仅出现一次。

8.6 铃声

MMAPI 可用于播放铃声，如 [第 A.13.1 节](#)，[第 A-30 页上的“Simple Tones”](#) 和 [第 A.13.2 节](#)，[第 A-31 页上的“Simple Player”](#) 中所演示。有几种铃声格式比较常用。您可以下载铃声或自己创建铃声。

8.6.1 下载铃声

铃声文件可以从多个 Internet 站点下载，包括以下站点：

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

8.6.2 铃声格式

本节提供了几种格式的样例

- RTTTL，铃声文本传输语言格式，在以下位置对这种格式进行了介绍：

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

- Nokia Composer

以下是 Nokia Composer 格式的贝多芬第九交响曲演奏：

```
16g1,16g1,16g1,4#d1,16f1,16f1,16f1,4d1,16g1,16g1,16g1,16#g1,  
16#g1,16#g1,16#g1,16g1,16#g2,16#g2,16#g2,4c2,16g1,16g1,16g1,  
16d1,16#g1,16#g1,16#g1, 16g1,16f2,16f2,16f2,4d2
```

- Ericsson Composer

贝多芬 G 大调小步舞曲：

```
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 p b f G p a e F
```

贝多芬第九交响曲旋律：

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

- Siemens Composer 格式

《G 型神探》主题曲：

```
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

贝多芬第九交响曲：

```
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

贝多芬第九交响曲：

```
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

贝多芬第九交响曲：

```
444*****444*****444*****111#*****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

本章简要概述了如何使用图形内容。Sun Java™ Wireless Toolkit for CLDC 提供了三种 API，它们为交互式 2D 和 3D 图形提供了全面的功能。

- Mobile 3D Graphics API for J2ME (JSR 184) 使用低级别 API 和高级别场景图形 API 提供了 3D 图形功能。本章提供了对使用 JSR 184 的简要概述以及基本指导。
 - Scalable 2D Vector Graphics API for J2ME (JSR 226) 支持呈现复杂的交互式 2D 内容。
 - Java Bindings for OpenGL® ES (JSR 239) 提供了 Java 语言接口以打开标准的 OpenGL® ES 图形 API。
-

9.1 使用 Mobile 3D Graphics API

JSR 184 是定义 Mobile 3D Graphics (Mobile 3D Graphics, M3G) API for J2ME 的规范。此 API 在一个适用于 CLDC/MIDP 设备的压缩包中提供了 3D 功能，并提供两种显示 3D 图形内容的方法。立刻模式 API 使应用程序可以直接创建和操纵 3D 元素。该 API 的顶层是一个场景图形 API，也称保留模式，此模式可以装入并显示事先设计的完整 3D 场景。应用程序可以随意使用最合适的方法，也可以随意使用保留模式和立刻模式 API 的组合。JSR 184 规范还为场景图形定义了一种文件格式 (.m3g)。

有关更多信息，请通过以下网址查询 JSR 184 规范：
<http://jcp.org/en/jsr/detail?id=184>。

JSR 184 提供了一个面向 CLDC/MIDP 设备的标准 API，支持新一代 3D 应用程序。而立刻模式 API 可以与 OpenGL ES 兼容，OpenGL ES 是面向 3D 图形的一个标准轻量级 API。有关 OpenGL ES 的更多信息，请参见 <http://khronos.org/>。

9.1.1 立刻模式

立刻模式适用于在算法上生成 3D 图形内容的应用程序，如科学视觉呈现或统计图。该应用程序可以直接创建并操纵 3D 对象。

有关立刻模式的示例，请参见 Demo3D 示例应用程序中的 Life3D MIDlet。

9.1.2 保留模式

大多数应用程序（尤其是游戏）会使用保留模式或场景图形 API。在这种情况下，图形设计师或美工会使用 3D 建模软件创建场景图形。场景图形以 JSR 184 文件格式保存。场景图形文件与应用程序捆绑在一起。在运行时，应用程序使用场景图形 API 来装入和显示文件。

应用程序可操纵已装入场景图形的某些部分，以创建角色动画或其他效果。基本策略就是尽可能多使用建模软件。在运行时，应用程序可抓取并操纵场景图形的某些部分，其中也可以包括动画或其他效果的路径。

有关保留模式的示例，请参见 Demo3D 示例应用程序中的 `retainedmode MIDlet`。

9.1.3 以质量博取速度

MIDP 开发面临的挑战之一就是一般设备的环境限制。与台式机相比，MIDP 设备处理器的处理速度较慢，内存也较小。这些问题同样也影响到 3D 图形处理区域。为了适合大部分实现，JSR 184 规范提供了各种机制以尽可能高效地显示 3D 场景。

一种方法就是范围界定，通过这种技术，您可以告诉 3D 图形实现何时对象之间将不发生交互。例如，如果您定义了一个房间场景图形，您可以使用范围界定，指定地下室的光线并不影响一层起居室的外观。通过减少显示场景所需的运算量，范围界定使实现工作变得更加简单。

但是一般来说，提高 3D 场景渲染速度的最好办法是在质量上做出一些让步。Mobile 3D Graphics API 包含渲染提示，应用程序可以提示实现如何在质量上做出让步以提高渲染速度。

9.1.4 创建 Mobile 3D 图形内容

大多数 Mobile 3D 应用程序使用资源文件中的场景图形来描述对象、场景和角色。场景图形通常并不是由编程人员来创建，而是由图形设计师或美工使用标准 3D 建模工具来创建。

有些供应商提供了用于创作内容以及将文件转换为 JSR 184 格式的工具。Superscape (<http://superscape.com/>) 就是这样一家供应商。

由于使用立刻模式 API 在应用程序中创建和操纵 3D 图形内容相对比较困难，因此大多数应用程序都更多地依赖某个场景图形文件。如果在设计过程中尽可能使用场景图形文件，在运行时应用程序的工作就简单多了。

9.2

渲染可伸缩矢量图形内容

可伸缩矢量图形 (Scalable Vector Graphics, SVG) 是由万维网联盟定义的标准。它是用于描述丰富的交互式 2D 图形的 XML 语法。

Sun Java™ Wireless Toolkit for CLDC 仿真器支持 Scalable 2D Vector Graphics API for J2ME (JSR 226)。JSR 226 是用于装入、操纵、渲染和播放 SVG 内容的 Java ME API。SVG Tiny 是简洁而强大的 XML 格式，用于描述丰富的交互式动画 2D 内容。

当可以使用文本编辑器生成 SVG 内容时，大多数用户更愿意使用创作工具。有以下三种可能性：

- **BeatWare Mobile Designer -**
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 内容的 Java ME 应用程序可以创建适合用户显示器的显示分辨率和波形因数的图形效果。

可以通过两种方式以动画显示 SVG 图像。一种是使用陈述性动画。另一种是通过 API 调用重复地修改 SVG 图像参数（例如颜色或位置）。第 A.21.3 节，第 A-48 页上的“播放 SVG 动画”说明了此陈述性动画。

9.3

OpenGL® ES 概述

JSR 239 定义用于两个 API (OpenGL® for Embedded Systems [OpenGL® ES] 和 EGL) 的 Java 编程语言绑定。OpenGL® ES 是用于 3D 图形的标准 API，是 OpenGL® 的子集，可以在台式机上普及。EGL 是标准平台接口层。OpenGL® ES 和 EGL 均由 Khronos Group (<http://khronos.org/opengles/>) 开发。

OpenGL® 是适用于访问硬件加速 3D 图形的低级别图形库，而 JSR 184（它是面向对象的）要求高级别功能。需要开发 OpenGL ES Demo 样例项目代码。

使用 PIM 和 FileConnection API

Sun Java™ Wireless Toolkit for CLDC 支持 PDA Optional Packages (PDAP) for the J2ME Platform (JSR 75)。JSR 75 包括两个独立的 API:

- FileConnection 可选包允许 MIDlet 访问本地设备文件系统。
- 个人信息管理 (Personal Information Management, PIM) 可选包包括用于操纵联系人列表 (通讯录)、日历以及任务列表的 API。

本章介绍 Sun Java™ Wireless Toolkit for CLDC 如何实现 FileConnection 和 PIM API。

10.1 FileConnection API

在真实设备上, FileConnection API 通常会提供对存储在设备内存或内存卡上文件的访问权限。

在 Sun Java™ Wireless Toolkit for CLDC 仿真器中, FileConnection API 允许 MIDlet 访问存储在台式机硬盘中的文件。

可以使用存储在 `workdir\appdb\skin\filesystem` 的子目录下的 FileConnection 访问这些文件。例如, DefaultColorPhone 仿真器样机附带了一个已安装的名为 `root1` 的根目录, 该目录包含一个名为 `Readme` 的文件和一个名为 `photos` 的空目录。文件的完整路径为:

对于 Windows 系统: `workdir\appdb\skin\filesystem\root1\photos`

对于 Linux 系统: `workdir/appdb/skin/filesystem/root1/photos`

注 – 如果同时运行同一个仿真器样机的多个实例，Sun Java™ Wireless Toolkit for CLDC 将为每个实例生成唯一的文件路径。例如，第一个目录被命名为 DefaultColorPhone，第二个实例被命名为 DefaultColorPhone1。

filesystem 的每个子目录称为 *root*。Sun Java™ Wireless Toolkit for CLDC 提供了根管理机制。当仿真器运行时，从仿真器窗口菜单中选择 "MIDlet" > “外部事件”。将显示用于添加和删除根的实用程序窗口。

图 10-1 管理文件系统根目录



使用 FileConnection API，应用程序可以访问已安装的根目录及其内容。

要添加新的根目录，请单击“安装新项目”，然后填写目录名。要使 FileConnection API 不能访问某个目录，请在列表中选中该目录，然后单击“卸载”。

10.2 PIM API

Sun Java™ Wireless Toolkit for CLDC 仿真器将联系人、日历以及任务信息以标准文件格式存储在您的台式机硬盘上。所有信息都存储在 `workdir\appdb\skin\pim` 中。该目录由所有运行中的仿真器共享。列表分别存储在 `contacts`、`events` 和 `todo` 目录的子目录中。例如，名为 `Contacts` 的联系人列表包含在：

对于 Windows 系统: `workdir\appdb\skin\pim\contacts\Contacts`

对于 Linux 系统: `workdir/appdb/skin/pim/contacts/Contacts`

在列表目录内部，各项都存储为 vCard (.vcs) 或 vCalendar (.vcf) 格式（请参见 <http://www.imc.org/pdi/>）。联系人存储为 vCard 格式，而日历和任务项都存储为 vCalendar 格式。

使用蓝牙和 OBEX API

Sun Java™ Wireless Toolkit for CLDC 仿真器支持 Java APIs for Bluetooth (JSR 82)。该仿真器完全符合 1.1 版规范，该规范描述了与推送注册表的集成。JSR 82 包括两个独立的 API：

- 蓝牙 API 提供蓝牙无线网络接口，包括设备搜索和数据交换。
- OBEX API 允许应用程序在蓝牙或其他通讯信道中使用对象交换 (Object Exchange, OBEX) 协议。

本章介绍 Sun Java™ Wireless Toolkit for CLDC 实现蓝牙和 OBEX API 的方法。

11.1 蓝牙模拟环境

用户可使用 Sun Java™ Wireless Toolkit for CLDC 仿真器在没有实际蓝牙硬件的情况下，开发并测试使用蓝牙的应用程序。该工具包模拟运行仿真器的蓝牙环境。多个仿真器实例之间能通过蓝牙 API 互相搜索并交换数据。

有关示例，请参见附录 A 中的 `BluetoothDemo` 文档。

11.2 OBEX 红外传输

Sun Java™ Wireless Toolkit for CLDC 通过模拟的蓝牙和红外连接来实现 OBEX 传输。模拟红外连接遵循 IrDA 标准（请参见 <http://www.irda.org/>）。可以在多个运行的仿真器之间进行模拟的红外传输。

11.3 设置 OBEX 和蓝牙首选项

用户可在 Sun Java™ Wireless Toolkit for CLDC 中配置蓝牙和 OBEX 模拟环境。选择“编辑” > “首选项”，然后选择“蓝牙 /OBEX”以显示以下窗口。

图 11-1 蓝牙和 OBEX 首选项



11.3.1 OBEX 首选项

使用 IrDA 的设备在真实世界中通过侦听来搜索其他设备。用户可使用首选项窗口 "IrDA OBEX" 部分的“搜索超时”字段来配置 Sun Java™ Wireless Toolkit for CLDC 仿真器搜索其他设备需等待的时间。输入以毫秒为单位的值。

在 API 级别，搜索超时值决定在调用返回或抛出异常之前，调用 `Connector.open("irdaobex://discover...")` 所阻塞的时间。

最大包长度将影响仿真器之间每个包中发送的数据量。包的长度越短，所需的包数就越多，等待处理的包也就越多。

11.3.2 蓝牙内部特性

在首选项窗口的“蓝牙”部分中，“设备搜索超时”指尝试查找模拟蓝牙环境中的其他设备时仿真器所等待的时间，以毫秒为单位。

“下一个仿真器的蓝牙地址”是指将指定给第一个仿真器实例的蓝牙地址。仿真器的后续实例将收到自动生成的地址。

11.3.3 蓝牙系统特性

首选项窗口“蓝牙”部分的“系统特性”选项卡包含可使用 `javax.bluetooth.LocalDevice` 中的 `getProperty()` 方法在应用程序中检索的特性。

JSR 82 规范对蓝牙特性进行了全面介绍。

11.3.4 蓝牙 BCC 特性

蓝牙控制中心 (Bluetooth Control Center, BCC) 用于控制蓝牙设置。有些设备可能会提供 GUI (图形用户界面) 来自定义蓝牙设置。在 Sun Java™ Wireless Toolkit for CLDC 中, BCC 是使用“蓝牙”首选项的“BBC 特性”选项卡配置的。特性如下:

表 11-1 BCC 特性

| 特性 | 描述 |
|--------|--|
| 启用蓝牙支持 | 如果禁用该特性, <code>LocalDevice.getLocalDevice()</code> 会抛出 <code>BluetoothStateException</code> , 并且无法创建任何连接。这用于测试支持 JSR 82 但可能已关闭蓝牙功能的设备上应用程序的性能。 |
| 设备可搜索 | 指示其他仿真器是否可搜索到该仿真器。 |
| 友好名称 | 模拟蓝牙环境中用户可读的仿真器名称。如果名称保留为空, 则仿真器不支持“友好名称”功能。 |
| 加密 | 该特性决定连接加密是否受支持 (支持则为 <code>on</code> , 不支持则为 <code>off</code>)。此外, <code>force</code> 设置表示所有连接均必须经过加密。有关详细信息, 请参见 <code>RemoteDevice</code> 的 <code>encrypt()</code> 方法的文档。 |
| 授权 | 与“加密”特性类似。请参见 <code>RemoteDevice</code> 的 <code>authorize()</code> 方法。 |
| 验证 | 与“加密”和“授权”特性类似。请参见 <code>RemoteDevice</code> 的 <code>authenticate()</code> 方法。 |

使用 Web 服务

Sun Java™ Wireless Toolkit for CLDC 仿真器支持 J2ME Web 服务规范 (JSR 172)。JSR 172 提供用于通过移动应用程序访问 Web 服务的 API。它还包括用于解析 XML 文档的 API。

Sun Java™ Wireless Toolkit for CLDC 提供了一个桩模块生成器，可自动创建用于访问 Web 服务的源代码。要进入桩模块生成器，请选择“文件”>“实用程序”。单击“桩模块生成器”，然后按“启动”。

图 12-1 Web 服务桩模块生成器



在“WSDL 文件名或 URL”字段中，输入要访问的 Web 服务的 Web 服务描述语言 (Webservices Description Language, WSDL) 文件的路径。“输出路径”表明要放置桩模块文件的位置。“输出包”表明桩模块文件的 Java 编程语言包的名称。最后，选择是要生成 CLDC 1.0 还是 CLDC 1.1 桩模块。

按“确定”生成桩模块文件。

使用 Location API

JSR 179 Location API 使应用程序可以使用设备的定位功能。例如，包含全球定位系统 (Global Positioning System, GPS) 硬件的某些设备，以及可以接收来自无线网络定位信息的其他设备。Location API 提供了定位信息的标准接口，而不考虑基础技术。

在 Location API 中，**位置提供者**可封装定位方法并提供有关设备位置的信息。应用程序通过指定所需的标准（例如所需的精度和响应时间）请求提供者。如果相应的实现可用，则应用程序可以使用它获取有关设备物理位置的信息。

Sun Java™ Wireless Toolkit for CLDC 包括模拟的位置提供者。您可以使用仿真器的“外部事件”窗口指定仿真器认为其应当所处的位置。另外，您可以配置提供者本身的属性，并可以管理路标的数据库。

13.1 设置仿真器运行时的位置

您可以指定仿真器运行时的模拟位置。要执行此操作，请从仿真器窗口的菜单中选择 "MIDlet" > “外部事件”。单击“位置”选项卡。请参见图 13-1。

图 13-1 控制仿真器中的位置



在选项卡的“位置”区域中，您可以填写纬度、经度、海拔高度、速度和方向的值。使用 Location API 的应用程序可以检索这些值作为仿真器的位置。

要进行更详细的测试，您可以设置随时间描述运动的位置脚本。位置脚本是由位置（称为航点）以及相关时间列表构成的 XML 文件。Sun Java™ Wireless Toolkit for CLDC 通过在位置脚本中的点之间插入新值确定仿真器的当前位置。例如，以下是一个简单的位置脚本，它指定了起点 (`time="0"`) 并在十秒后移动到一个新点。

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

海拔高度测量以米为单位，时间值以毫秒为单位。

使用文本编辑器可以创建位置脚本。您可以通过按下“脚本”字段旁边的“浏览”按钮将其装入到“外部事件”窗口。正下方是一些控件，用于播放、暂停、停止和移动到位置脚本的起点和终点。您也可以将时间滑尺拖动到特定点。

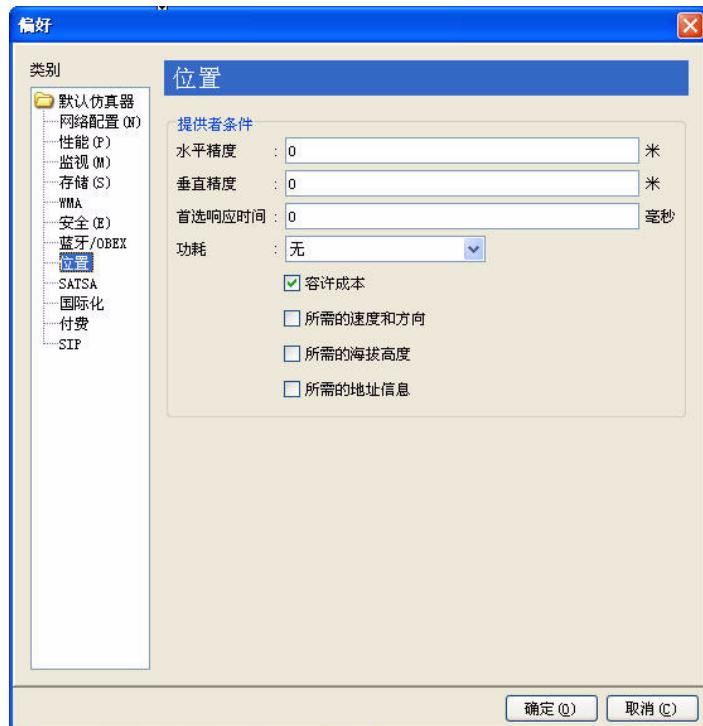
某些设备也能测量其方位。要使这种信息可用于您的应用程序，请将“方位”框中的“省 / 自治区 / 直辖市”字段更改为“受支持”并填写方位角、垂直倾斜角和旋转角的值。“磁方位”复选框指示方位角和垂直倾斜角测量是与地球的磁场相关还是与真北和重力相关。

要测试您的应用程序如何处理意外的情况，请尝试将“位置提供者”框中的“省 / 自治区 / 直辖市”字段更改为“暂时不可用”或“停止服务”。当您的应用程序尝试检索仿真器的位置时，会抛出异常并且您会看到应用程序是如何进行响应的。

13.2 配置位置提供者

您可以使用首选项配置 Sun Java™ Wireless Toolkit for CLDC 位置提供者的属性。在用户界面中选择“编辑”>“首选项”，然后单击“位置”。

图 13-2 配置位置提供者

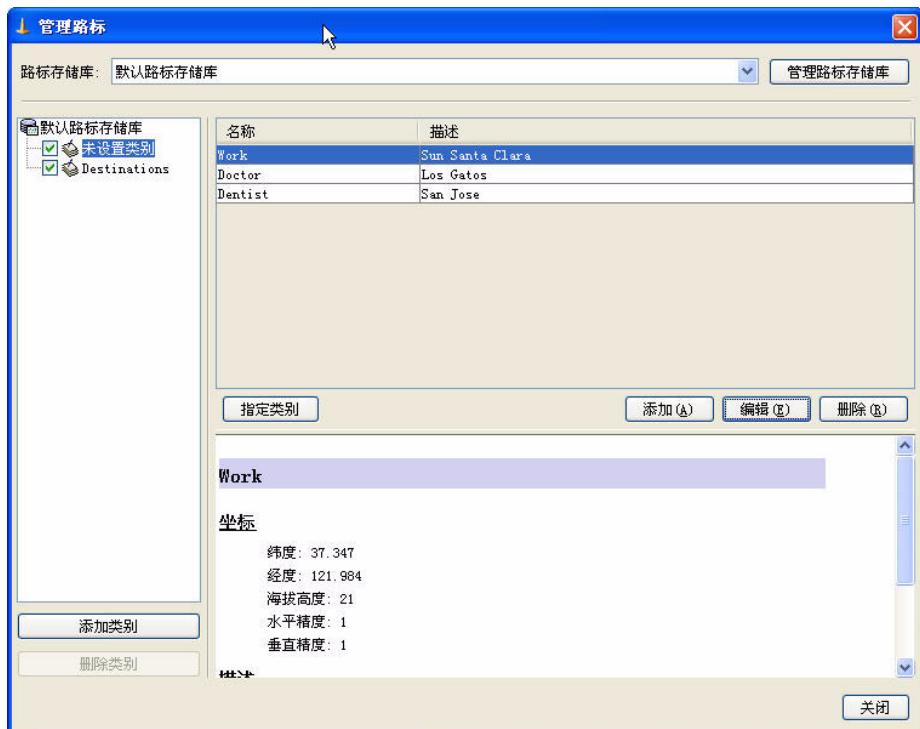


使用“位置”选项卡中的字段，您可以指定工具包的内置位置提供者的属性。在与 Criteria 类应用程序相对应的首选项中指定的属性用于请求位置提供者。

13.3 设置路标

正如许多真实设备一样，Sun Java™ Wireless Toolkit for CLDC 仿真器包括路标存储库系统。路标存储库是具有相关联名称和其他信息的位置的集合。要管理路标存储库，请从菜单中选择“文件”>“实用程序”，并选择“管理路标”然后按“启动”。

图 13-3 路标管理器



路标管理器显示单个路标存储库的内容。JSR 179 要求最少使用一个路标存储库，该存储库称为默认存储库。要选择其他路标存储库或创建新的路标存储库，请从窗口顶部的“路标存储库”组合框中进行选择。

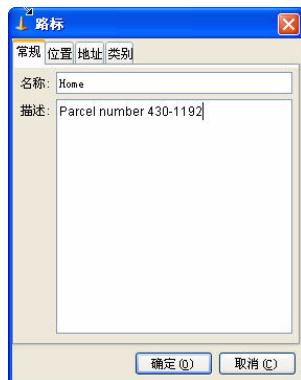
您可以通过单击“管理路标存储库”按钮添加或删除路标存储库。无法对路标存储库进行重命名。

路标可以与特定于路标存储库的类别相关联。当前路标存储库的类别显示在窗口的左窗格中。您可以使用列表底部的按钮添加或删除类别。如果希望仅显示匹配的路标，请选中一个或多个类别。您也可以选中“未设置类别”以显示无关联类别的路标。

路标管理器的右窗格列出了当前路标存储库中的路标。单击路标以查看右窗格底部列出的详细信息。

要添加新路标，请单击“添加”并相应地填写字段。单击“编辑”以更改当前选定的路标。最后，按“删除”以删除当前选定的路标。

图 13-4 添加或编辑路标



您也可以使用主窗口中的“指定类别”指定路标的类别。

使用 SATSA

安全和信任服务 API (Security and Trust Services API, SATSA) 使小型设备上运行的应用程序具备智能卡访问和加密功能。JSR 177 (SATSA 规范) 定义了四种不同的 API, 作为可选包:

- **SATSA-APDU**——允许应用程序使用低级协议与智能卡应用程序进行通讯。
- **SATSA-JCRMI**——使用远程对象协议, 为与智能卡应用程序进行通讯提供备用方法。
- **SATSA-PKI**——允许应用程序使用智能卡以数字方式签名数据并管理用户证书。
- **SATSA-CRYPTO**——一般用途的加密 API, 可以支持消息摘要、数字签名以及加密算法。

Sun JavaTM Wireless Toolkit for CLDC 仿真器完全支持 SATSA。本章介绍如何使用 Sun JavaTM Wireless Toolkit for CLDC 在您自己的应用程序中使用 SATSA。

有关 SATSA 以及在小型设备中使用智能卡的一般介绍, 请参见 SATSA Developer's Guide, 可从以下位置获得:

<http://java.sun.com/j2me/docs/satsta-dg/>。

Sun JavaTM Wireless Toolkit for CLDC 中包含 Java Card 平台模拟器, 用户可以使用它模拟 Sun JavaTM Wireless Toolkit for CLDC 仿真器插槽中的智能卡。可以在以下位置找到 Java Card 平台模拟器。

对于 Windows 系统: toolkit\bin\cref.exe

对于 Linux 系统: toolkit/bin/cref

以下简称为 cref。

如果用户需要开发自己的 Java Card 应用程序, 请从

<http://java.sun.com/products/javacard/> 网站上下载 Java Card 开发工具包。

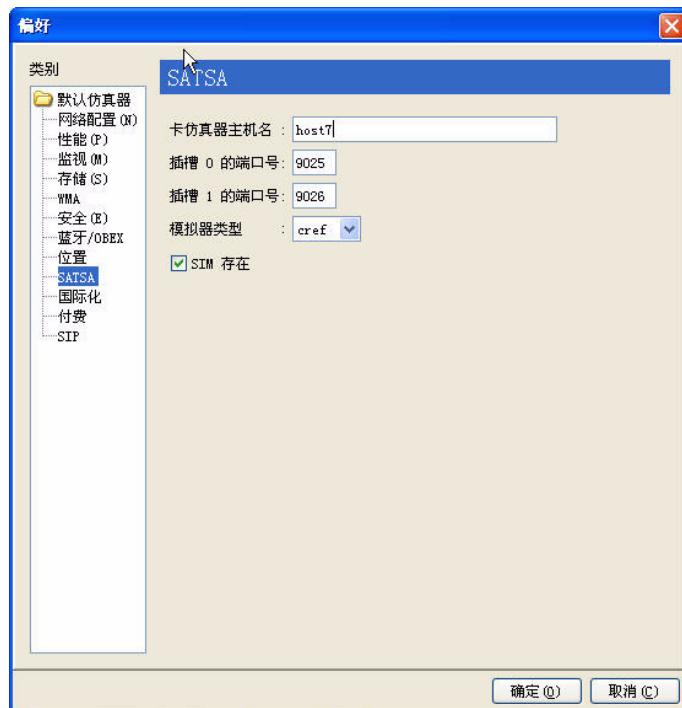
14.1 仿真器中的卡插槽

真实 SATSA 设备可能具有一个或多个可插入智能卡的插槽。使用 SATSA 与智能卡进行通讯的应用程序需指定一个插槽和卡应用程序。

由于 Sun Java™ Wireless Toolkit for CLDC 仿真器不是真实设备，因此不具有智能卡的物理插槽，而是使用套接字协议与智能卡应用程序进行通讯。套接字的另一端可以为智能卡仿真器或与真实智能卡硬件进行通讯的代理。

Sun Java™ Wireless Toolkit for CLDC 仿真器包含两个模拟的智能卡插槽。每个插槽均具有关联的套接字，表示用于与智能卡应用程序进行通讯的协议的一端。用户可为每个插槽设置套接字端口号。选择“编辑”>“首选项”，然后单击“SATSA”选项卡。插槽 0 的默认端口号为 9025，插槽 1 的默认端口号为 9026。

图 14-1 为智能卡插槽设置端口号



14.2

使用 Java Card 平台模拟器

通过 Sun Java™ Wireless Toolkit for CLDC 测试 SATSA 应用程序的基本步骤如下：

1. 使用 Java Card 平台应用程序启动 `cref`。
2. 启动 Sun Java™ Wireless Toolkit for CLDC 仿真器。

SATSA 应用程序在尝试与智能卡进行通讯时，会使用套接字连接与 `cref` 进行通讯。

因此，请务必确保所启动 `cref` 时所使用的端口号与在 Sun Java™ Wireless Toolkit for CLDC 的首选项中指定的其中一个插槽端口号相同。

例如，用户可使用以下命令行在具有预生成的内存映像的端口 9025 上运行 `cref`：

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

Sun Java™ Wireless Toolkit for CLDC 包含演示应用程序 Mohair，其演示了如何使用 SATSA。有关运行 Mohair 的详细指导，请参见[附录 A](#)。

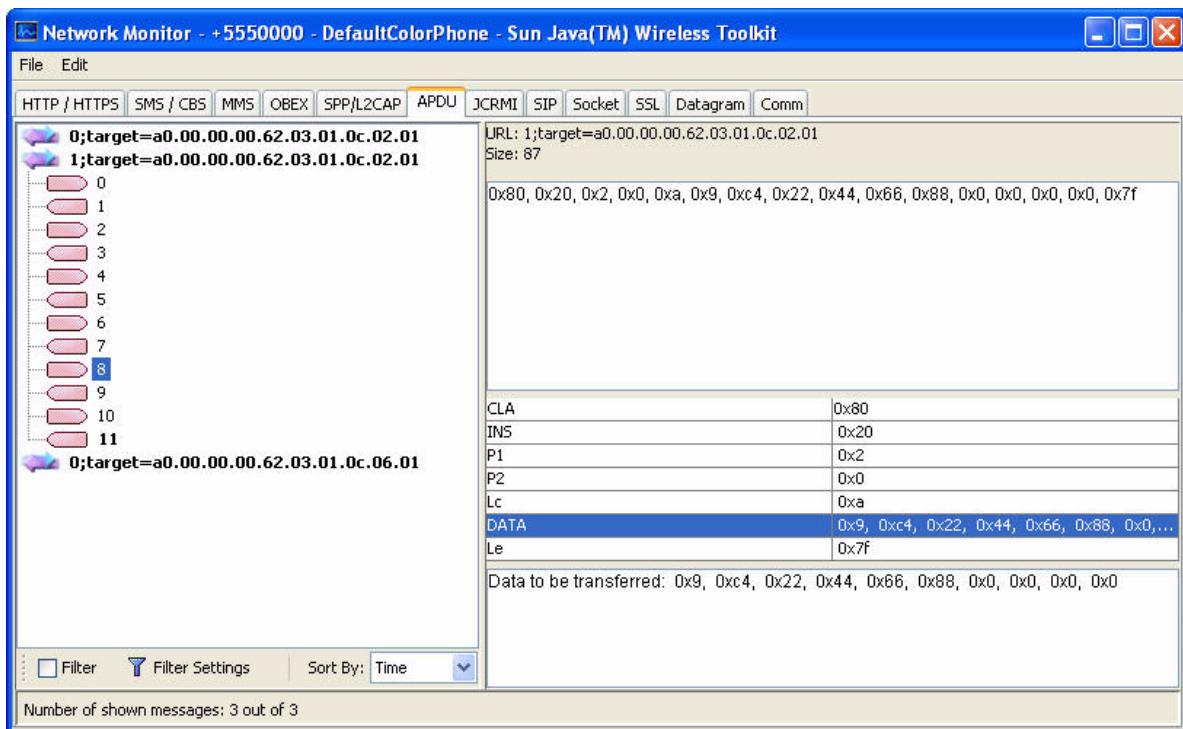
14.3

使用具有 SATSA 的网络监视器

Sun Java™ Wireless Toolkit for CLDC 可显示在网络监视器中与模拟的智能卡应用程序交换的数据。网络监视器显示在仿真器和智能卡模拟器之间交换的应用协议数据单元 (Application Protocol Data Units, APDU)。其还可以显示使用 Java Card 远程方法调用 (Java Card RMI) 协议交换的数据。网络监视器中的 "APDU" 和 "JCRMI" 选项卡显示了通过智能卡交换的数据。

网络监视器可以解析每个 APDU，并根据需要显示请求和响应中的字段。

图 14-2 在网络监视器中查看 APDU



14.4 调整访问控制

可在文本文件中指定访问控制权限和 PIN 属性。建立第一个 APDU 或 Java Card RMI 连接后，实现会读取位于 *workdir\appdb* 目录下 *acl_slot-number* 中的 ACL 和 PIN 数据。例如，插槽 0 的访问控制文件为 *workdir\appdb\acl_0*。如果文件不存在或包含错误，则将禁用该插槽的访问控制验证。

该文件可以包含有关 PIN 属性和应用程序权限的信息。

14.4.1 指定 PIN 属性

在访问控制文件中， PIN 属性通过 `pin_data` 记录表示。

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

14.4.2 指定应用程序权限

应用程序的权限在访问控制文件 (acf) 记录中进行定义。

```
acf AID f 由空格分隔的数字 {
    ace {
        root CA 名称
        ...
        apdu {
            由空格分隔的八个数字
            ...
        }
        ...
        jc.rmi {
            classes {
                类名
                ...
            }
            hashModifier 字符串
            methods {
                方法名称和签名
                ...
            }
        }
        ...
        pin_apdu {
            id number
        }
    }
}
```

```

        verify | change | disable | enable | unblock
        四个十六进制的数字
        ...
    }
    ...
    pin_jc.rmi {
        id number
        verify | change | disable | enable | unblock
        方法名称和签名
        ...
    }
    ...
}
...
}

```

acf 记录是一个访问控制文件。 **acf** 之后的 AID 用于标识应用程序。缺少 AID 表示该条目将应用于所有应用程序。 **acf** 记录可包含 **ace** 记录。如果没有 **ace** 记录，则该 **acf** 将限制对应用程序的访问。

ace 记录是一个访问控制条目。它可以包含 **root**、**apdu**、**jc.rmi**、**pin_apdu** 和 **pin_jc.rmi** 记录。

root 记录包含一个 CA 名称。如果 MIDlet 套件是使用由该 CA 颁发的证书进行授权的，则该 **ace** 将授予访问该 MIDlet 的权限。缺少 **root** 字段则表示 **ace** 将应用于所有标识的部分。一行描述一个主体。此行必须只包含字 **root** 以及主体名称，例如：

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

apdu 或 **jc.rmi** 记录介绍 APDU 或 Java Card RMI 权限。缺少权限记录则表示允许执行所有操作。

APDU 权限包含一组或多组八个十六进制值序列，由空格分隔。前四个字节描述 APDU 命令，其余四个字节为掩码，例如：

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

Java Card RMI 权限包含有关散列修饰符（可选）、类列表以及方法列表（可选）的信息。如果方法列表为空，则允许应用程序调用类列表中接口的所有远程方法，例如：

```

jc.rmi {
    classes {
        com.sun.javacard.samples.RMIDemo.Purse
    }
    hashModifier zzz
    methods {

```

```

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

```

所有的数字均为十六进制。制表符、空格符、回车符及换行符用作分隔符。符号 { 和 } 前后可省略分隔符。

`pin_apdu` 和 `pin_jcrmi` 记录包含 PIN 条目方法所必需的信息，这些信息为 PIN 标识符和 APDU 命令标题，或远程方法名称。

14.4.3 访问控制文件示例

```

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

    jc.rmi {
        classes {
            com.sun.javacard.samples.RMIDemo.Purse
        }
        hashModifier xxx
        methods {
            setAccountNumber([B)V
            getBalance()S
            credit(S)V
        }
    }
}
ace {
    jc.rmi {
        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 通过代理服务器、注册器以及网络监视器支持来支持 SIP API for J2ME (JSR 180)。

会话发起协议 (Session Initiation Protocol, SIP) 是通过 RFC 3261 定义的, 请访问 <http://www.ietf.org/rfc/rfc3261.txt>。

SIP 提供了一种标准方式, 供应用程序设置通讯。应用程序决定实际要发生的通讯。可将 SIP 用于设置即时消息传送、文本聊天、语音聊天、视频会议或其他类型的会话。

15.1 了解注册器和代理

SIP 注册器允许客户机应用程序将用户名和特定的网络地址进行关联。实际上, 注册过程即相当于表明用户在此。

实际上, SIP 代理服务器只是通向较大的代理服务器网络的入口点。会将到达一个代理的 SIP 消息路由到相应的目的地, 该目的地通常为其他代理服务器或某个端点, 例如台式机或移动设备。尽管可以在设备之间直接发送 SIP 消息, 但通常通过代理服务器路由 SIP 消息。

例如, 假设 Diggory 要与 Polly 进行视频会议, 而 Polly 正在途中, 则她的移动电话会向注册器发送消息, 注册器会将其姓名与移动电话的网络地址进行关联。当 Diggory 尝试设置与 Polly 的视频会议时, 他的应用程序将使用 SIP 询问注册器 Polly 当前的网络位置。

Sun Java™ Wireless Toolkit for CLDC 包含的 SIP 代理和注册器服务器十分简便, 用户可使用它们来测试使用 SIP API 的应用程序。用户也可以将工具包配置为使用外部代理服务器和注册器服务器。

15.2 SIP 设置

要调整 Sun Java™ Wireless Toolkit for CLDC 的 SIP 环境的设置，请选择“编辑”>“首选项”，然后单击“SIP”。

图 15-1 SIP 设置



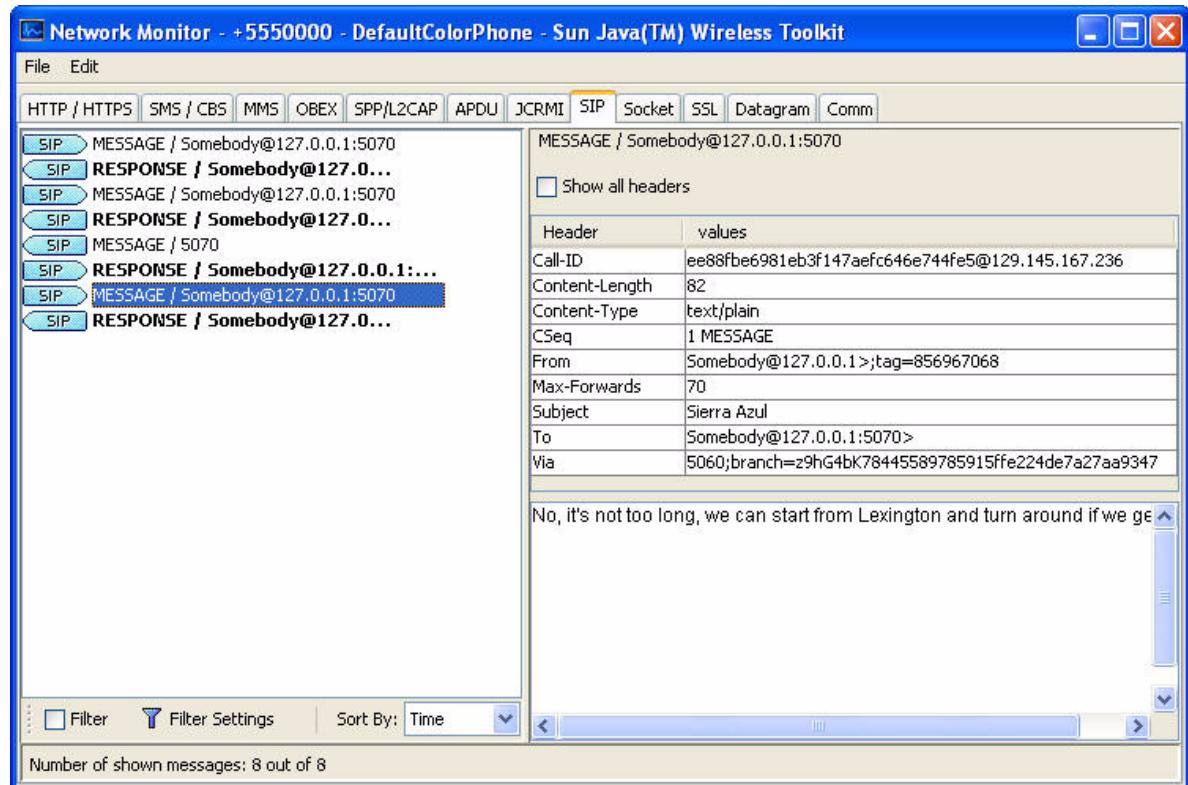
“显示名称”和“地址”字段分别用于设置系统特性 `sip.display.name` 和 `sip.address`。应用程序可将这些系统特性用作检索与设备关联的身份的标准方法。

15.3 网络监视器中的 SIP 通信流量

可通过网络监视器来记录使用 SIP API 发送和接收的网络数据。第 5 章中对网络监视器进行了全面介绍。用户可使用网络监视器跟踪从仿真器发送或由仿真器接收的 SIP 消息。

单击 "SIP" 选项卡可查看 SIP 消息。SIP 消息显示在网络监视器的左窗格中。单击某一消息可在右窗格中查看其详细信息。

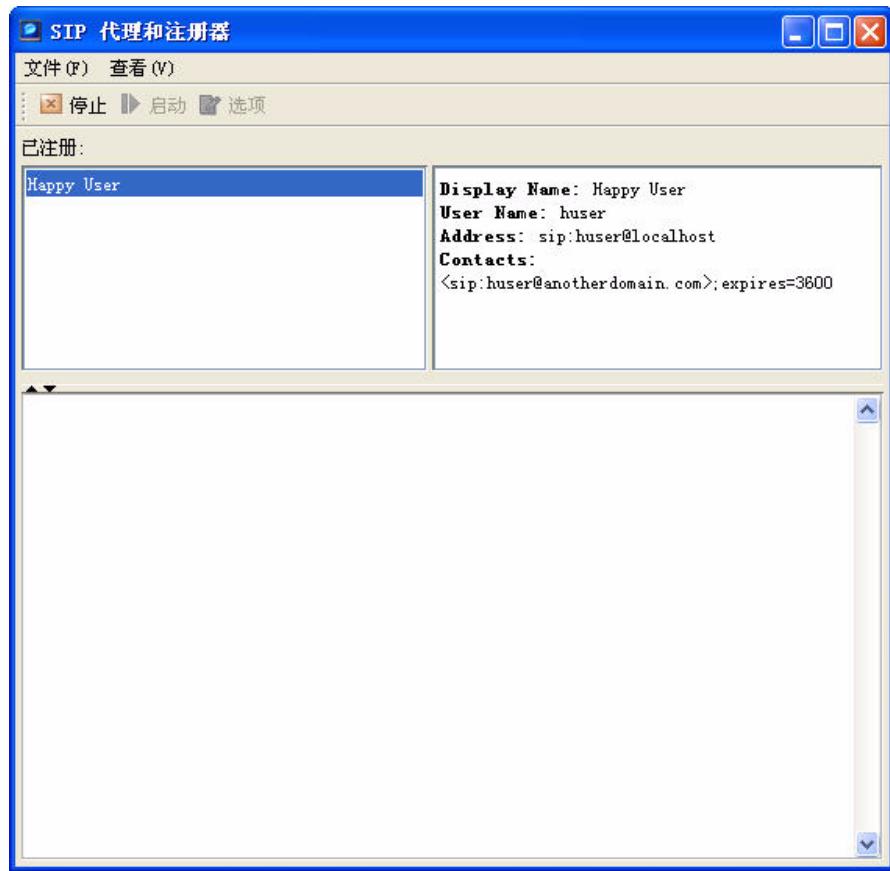
图 15-2 网络监视器中的 SIP 消息



15.4 SIP 代理服务器和注册器

Sun Java™ Wireless Toolkit for CLDC 提供了简便的 SIP 代理服务器和注册器，使用户可轻松创建 SIP 应用程序。要启动服务器，请选择“文件”>“实用程序”。从列表中选择“启动 SIP 服务器”，然后按“启动”。屏幕将显示“SIP 服务器控制台”窗口。

图 15-3 SIP 服务器控制台



要停止服务器，请单击“停止”。要启动服务器，请单击“启动”。

服务器运行时，左上方窗格中将显示注册器已知的所有用户。单击某个用户名可在右上方的窗格中看到有关该用户的详细信息。窗口的底部窗格为控制台，其中显示通过代理接收和发送的 SIP 消息。

用户可在服务器不运行时调整服务器选项。停止服务器，然后单击“选项”查看选项窗口。（请参见图 15-4）。

图 15-4 设置代理选项



在“代理”选项卡上，用户可以设置 IP 地址以及服务器侦听到的消息的端口。请注意，默认侦听点 5060 是 SIP 代理的已知端口。如果您在多用户环境下工作，有时其他用户可能会使用该端口，您可能会无意中连接到其他用户的 SIP 服务器实例（SIP 服务器不具有任何验证机制，可以自由地访问 TCP/IP 端口）。如果出现这种情况，您必须指定其他端口。

单击“添加”指定更多端口及其类型。选择某个端口，然后单击“删除”删除侦听端口。

选中“使用验证”可以强制连接的客户机在服务器端进行自行验证。所使用的方案为摘要验证，RFC 3261 的第 22.4 节中对其进行了介绍。SIP 的摘要验证与 HTTP 摘要验证相近。

在“注册器”选项卡中，可以设置注册器已知的用户和域。顶部列表包含启动 SIP 服务器时自动注册的 SIP 用户。您可以添加新用户，编辑现有用户，或删除用户。

此外，您还可以调整由该注册器管理的域的列表。按“添加”添加域，按“编辑”编辑现有域名，或按“删除”删除域。

图 15-5 设置注册器选项



使用 Payment API

JSR 229 (Payment API) 使应用程序可以代表其用户进行付费。通过付费适配器，Payment API 可支持不同的付费机制。实现 Payment API 的设备可具有一个或多个适配器。MIDlet 套件使用描述符属性指定它们可以使用的付费适配器的类型。

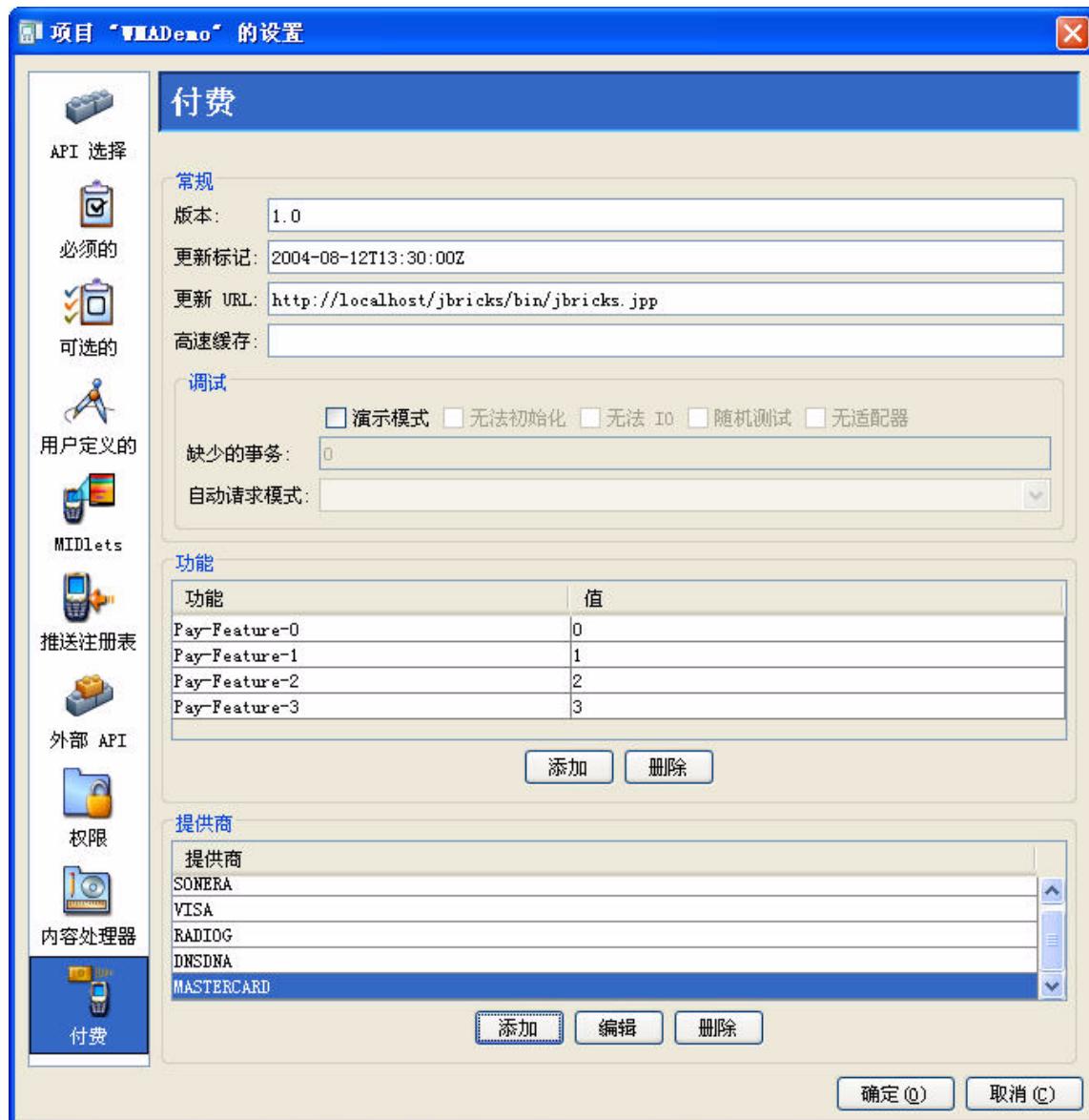
Sun Java™ Wireless Toolkit for CLDC 的仿真器通过可以模拟“特级短信价”的短信 (Premium Priced SMS, PPSMS) 和信用卡付费的示例付费适配器实现 Payment API。此外，使用工具包可以很容易设置 MIDlet 的描述符和 JAR 文件清单中的必要属性。最后，使用付费控制台可以轻松地跟踪通过应用程序所执行或尝试的付费。

由于 Payment API 与置备 (provisioning) 和外部设备付费机制紧密相关，并且付费仅可以在受信任的保护域中进行，因此请始终使用工具包的“通过 OTA 运行”机制测试和调试您的 Payment API 应用程序。有关详细信息，请参见[第 2 章](#)。

16.1 项目付费设置

要调整项目的付费属性，请单击“设置”并选择“付费”图标。

图 16-1 付费设置



这些字段和值在 JSR 229 (Payment API) 规范中进行了详细地介绍。

“常规”框包含有关使用中的 Payment API 版本以及付费更新位置的信息。要进行测试，您可以指定用于从项目目录中直接检索更新文件的 localhost URL（如屏幕快照所示）。

“调试”框包含在应用程序测试期间要使用的选项。每个选项都在 Payment API 规范中进行了介绍。

“功能”框列出了应用程序可以收费的功能。这些功能与为每个提供商列出的价格信息相对应。您可以使用“添加”和“删除”按钮修改功能列表。

“提供商”框列出了可以用于此应用程序的特定付费提供商。到达付费时间时，仿真器（或设备）会将一个可用的付费适配器与应用程序中列出的一个提供商进行匹配。您可以使用“添加”、“编辑”和“删除”按钮修改提供商列表。如果您添加了提供商或对提供商进行了编辑，会显示以下窗口。

图 16-2 编辑付费提供商



这些字段也在 Payment API 规范中进行了详细介绍。

对于每个已定义的付费功能，在“价格信息”框中均包含与其相对应的一行信息。要编辑价格标记的值，请在“值”列中双击相应的单元。

16.2 直接编辑付费属性

付费属性存储在扩展名为 .jpp 的付费更新文件中。有关完整的详细信息，请阅读规范。Sun Java™ Wireless Toolkit for CLDC 提供的实用程序使您可以独立于项目设置而轻松编辑付费更新文件。

要运行该实用程序，请选择“文件”>“实用程序”，并选择“付费编辑对话框”然后按“启动”。系统将提示您选择要编辑的付费更新文件。选择文件后，将显示一个外观与项目设置中的“付费”部分几乎相同的窗口。付费编辑实用程序中不包含调试设置。

编辑付费更新设置，并在完成后按“确定”。由于付费更新文件可以进行加密签名，因此工具包将显示可以用来对文件进行签名的密钥列表。选择要使用的密钥，然后按“对付费更新文件进行签名”。

16.3 付费首选项

要调整工具包的 Payment API 设置，请选择“编辑”>“首选项”，然后单击“付费”。

图 16-3 设置付费首选项



“移动电话国家 / 地区代码” (Mobile Country Code, MCC) 和 “移动电话网络代码” (Mobile Network Code, MNC) 与 PPSMS 付费提供商结合使用。通过 MCC 和 MNC 的组合可以标识设备的无线运营商。付费时，MCC 和 MNC 用于从项目设置的提供商列表中查找匹配的提供商。由于仿真器不是真实设备，因此您可以通过填写适当的 MCC 和 MNC 值模拟运营商。有关详细信息，请查询 Payment API 规范。

“过去的事务限制”用于确定在仿真器中所记录的先前事务的数目。这将会影响在外部事件窗口中显示的列表的长度（在下面进行了介绍）以及应用程序本身可以检索的先前事务的数目。

最后，“控制台电话号码”用于确定付费控制台的模拟电话号码（随后将对其进行介绍）。

16.4 查看事务历史

仿真器可以像真实设备一样跟踪付费事务。要查看事务的历史，请从仿真器窗口的菜单中选择 "MIDlet" > “外部事件”。单击“付费事务”选项卡。

图 16-4 查看付费事务



单击“刷新”以在进行更多付费后更新列表。选择事务并单击“详细信息”以在单独的窗口中查看所有详细信息。

外部事件窗口仅显示仿真器在通过 OTA 运行时所执行的事务。尽管不使用 OTA 也可以完成事务，但不会显示此类事务。要获得对付费最真实的模拟效果，请始终使用“通过 OTA 运行”测试应用程序。

16.5 监视付费

Sun Java™ Wireless Toolkit for CLDC 提供的付费控制台使您可以轻松看到付费通过示例付费适配器的过程。要启动“付费服务提供商控制台”，请选择“文件”>“实用程序”>“付费控制台”。

图 16-5 付费控制台



此外，您可以使用网络监视器查看事务。信用卡事务可以使用 HTTPS 执行，而 PPSMS 事务可以使用 SMS 执行。有关网络监视器的完整说明，请参见第 5 章。

使用 Mobile Internationalization API

JSR 238 (Mobile Internationalization API) 用于要以多种语言显示并在多个国家 / 地区使用的应用程序。国家（或地区）和语言的组合是语言环境。

JSR 238 的核心概念是资源，它是适用于特定语言环境的字符串、图像或其他对象。例如，将要在欧洲发布的应用程序可能包含以下用户的资源：居住在意大利讲意大利语的用户、居住在瑞士讲意大利语的用户、居住在西班牙讲西班牙语的用户以及居住在葡萄牙讲西班牙语的用户等等。

资源存储在 JSR 238 中所定义格式的文件中。这些资源文件作为 MIDlet 套件 JAR 文件的一部分捆绑在一起。Sun Java™ Wireless Toolkit for CLDC 提供了可以简化创建和维护资源文件工作的资源管理器。

17.1 设置仿真器的语言环境

设备的语言环境包含在系统属性 `microedition.locale` 中。您可以通过选择“编辑”>“首选项”，然后选择“国际化”更改仿真器的语言环境。从组合框中选择语言环境或直接键入语言环境。

17.2 查看应用程序资源

要启动资源管理器，请选择“文件”>“实用程序”。选择“国际化资源管理器”，然后单击“启动”。

图 17-1 资源管理器



首先，从“项目”菜单中选择项目。选定项目的所有资源将显示在窗口的其余部分。如果这是您第一次查看资源管理器，请查看 i18nDemo 演示应用程序中的资源，其中包含许多有趣的资源。

您可以单击 "+" 号展开目录或单击 "-" 号收拢目录。

通过在目录或资源文件上单击鼠标右键，几乎可以执行所有其他操作。

窗口的顶部窗格显示了应用程序中资源文件的层次结构，而底部窗格则显示了资源文件的内容。在图 17-1 中，显示了顶层 common 资源文件的内容。

17.3 使用语言环境

语言环境由顶层 `global` 目录下的目录表示。语言环境目录包括资源文件，这些资源文件中保存了可以用于应用程序的实际资源。

语言环境由标准语言和国家 / 地区代码表示，如 MIDP 2.0 规范中所述。例如 `pt-BR` 表示居住在巴西讲葡萄牙语的用户。

要添加语言环境目录，请在顶层 `global` 目录上单击鼠标右键，然后选择“添加语言环境”。从组合框中选择语言环境或直接键入语言环境，然后单击“确定”。

要重命名语言环境，请在语言环境目录上单击鼠标右键，然后选择“重命名”。

要删除语言环境及其包含的所有资源文件，请在语言环境目录上单击鼠标右键，然后选择“删除”。

17.4 使用资源文件

资源文件可以是全局资源文件（位于顶层），也可以位于语言环境目录中。要创建新的全局资源文件，请在顶层 `global` 目录上单击鼠标右键，然后选择“添加新资源文件”。选择该文件的名称。

通过在文件上单击鼠标右键，然后选择“重命名”对资源文件进行重命名。

通过在文件上单击鼠标右键，然后选择“删除”删除资源文件。

您可以复制、剪切和粘贴整个资源文件。在文件上单击鼠标右键，然后选择“复制”或“剪切”。然后在语言环境目录（或顶层 `global`）上单击鼠标右键并选择“粘贴”。

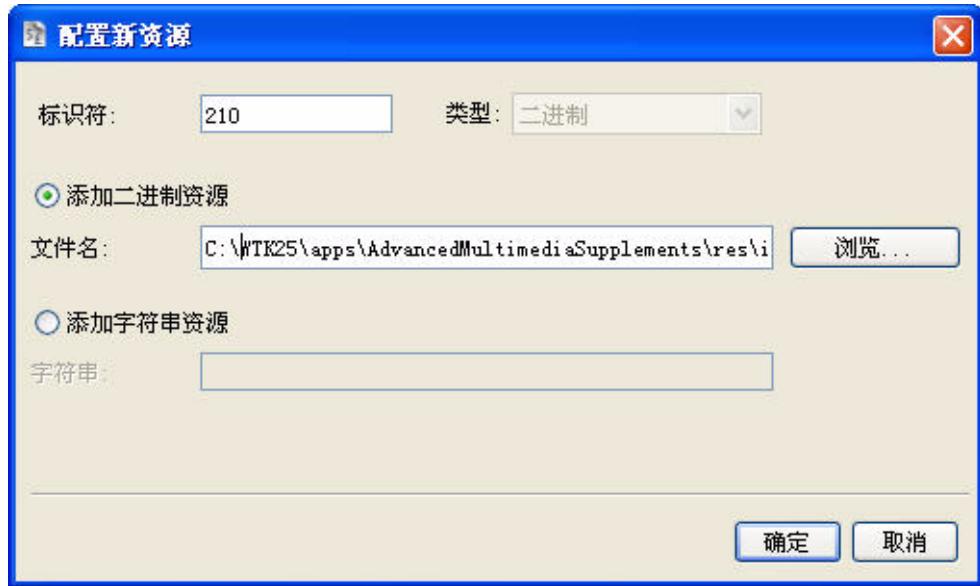
17.5 使用资源

在资源管理器窗口顶部窗格中的资源文件上单击，以在底部窗格中查看其内容。

要添加新的文本资源，请单击“添加”并选择“添加字符串资源”。将自动指定标识符，但您可以输入其他标识符（如果希望）。单击“确定”。

要添加图像或其他类型的数据，请单击“添加”并选择“添加二进制资源”。您可以浏览到所希望添加的任何文件，并且，您可以更改标识符（如果希望）。单击“确定”添加资源。

图 17-2 添加图像资源



要编辑资源，请双击该资源。您可以键入字符串资源的新值。如果双击文件资源，您可以选择其他文件。

演示应用程序

本附录介绍 Sun Java™ Wireless Toolkit for CLDC 所捆绑的演示应用程序。

A.1 概述

Sun Java™ Wireless Toolkit for CLDC 中附带的演示应用程序重点说明了仿真器支持的一些技术和 API。这些演示程序的目的是为了让用户对仿真器的 API 功能以及工具包的增强功能有一个大致了解。

表 A-1 中列出了该版本中包括的所有演示应用程序。

大多数演示程序的运行十分简单。[第 A.2 节](#), [第 A-3 页上的“一般说明”](#) 中包含有关运行大多数演示程序的说明。包含附加文档资料的演示程序在表 A-1 中均有链接。如果没有链接，则表明演示程序较为简单（或有自己的说明），一般的说明就足够了。

各个演示应用程序的源代码可在 *toolkit/apps* 目录中找到。子目录包含项目，并且每个项目均有包含 Java 编程语言源代码的 *src* 目录。例如，在 Windows 上，如果将工具包安装在 C:\WTK2.5.2 中，则 WMADemo 中 SMS 发件人 MIDlet (*example.sms.SMSSend*) 的源代码将位于 C:\WTK2.5.2\apps\WMADemo\src\example\sms\SMSSend.java 中。如[第 1.1.2 节](#), [第 1-2 页上的“工作目录文件”](#) 中所述，在打开项目时，会将源代码复制到您的 *workdir/apps* 目录中。

表 A-1 演示应用程序

| 演示程序 | API | 描述 | 特别说明 |
|--------------------------------|---|---------------------------|---------------------|
| AdvancedMultimedia Supplements | JSR 234 Advanced Multimedia Supplements | 展示 3D 音频、混响、图像处理和相机控制。 | A.3 |
| Audiodemo | MMAPI 1.1 | 演示音频功能，包括混音和播放与动画配合播放的音频。 | |
| BluetoothDemo | JSR 82 Bluetooth | 演示如何使用蓝牙进行设备搜索和数据交换。 | A.4 |

表 A-1 演示应用程序 (续)

| 演示程序 | API | 描述 | 特别说明 |
|------------------|---|--|------|
| CHAPIDemo | JSR 211 CHAPI | 一种内容查看器，也使用 MediaHandler。 | A.5 |
| CityGuide | JSR 179 Location API | 显示可显示当前位置路标的城市地图。 | A.6 |
| Demos | MIDP 2.0 | 包括各种示例：动画、颜色、网络、财务及其他。 | A.7 |
| Demo3D | JSR 184 Mobile 3D Graphics | 包含演示如何在立刻模式与保留模式下使用 3D 图形的 MIDlet。 | A.8 |
| FPDemo | CLDC 1.1 | 简单的浮点计算器。 | |
| Games | MIDP 2.0 | 包括 TilePuzzle、WormGame 和 PushPuzzle。 | |
| GoSIP | JSR 180 SIP | 演示使用 SIP 和 SIP 服务器时如何设置聊天 | A.9 |
| i18nDemo | JSR 238 Mobile Internationalization API | 包括字符串排序、数字格式化和短语翻译器。 | A.10 |
| JBricks | JSR 229 Payment API | 使用 Payment API 购买额外生命或级别的游戏。 | A.11 |
| JSR172Demo | Web 服务 | 演示如何使用 JSR 172 API 进行从 MIDlet 到 Web 服务的连接。 | A.12 |
| MobileMediaAPI | MMAPI 1.1 | 演示 MMAPI 功能，包括铃声序列、MIDI 回放、采样音频回放和视频。 | A.13 |
| NetworkDemo | MIDP 2.0 | 显示如何使用数据报和串行连接。 | A.14 |
| ObexDemo | JSR 82 OBEX | 演示如何使用 OBEX 通过 IrDA 传输数据。 | A.15 |
| OpenGL® ES Demo | JSR 239 OpenGL® ES | 显示如何使用 OpenGL® ES 创建 3D 图形。 | |
| PDAPDemo | JSR 75 PIM 和 FileConnection | 显示如何操纵联系人、日历项以及任务项。演示如何访问本地文件。 | A.16 |
| Photoalbum | MIDP 2.0 | 演示各种图像格式。 | |
| SATSADEMO | JSR 177 SATSA | 演示与智能卡进行通讯以及 SATSA 的其他功能。 | A.17 |
| SATSAJCRMIDemo | JSR 177 SATSA | 显示如何使用 SATSA-Java Card RMI。 | A.18 |
| SIPDemo | JSR 180 SIP | 使用 SIP 交换简单消息。 | A.19 |
| SnapMobileSample | | 演示连接的社区游戏功能。 | |

表 A-1 演示应用程序（续）

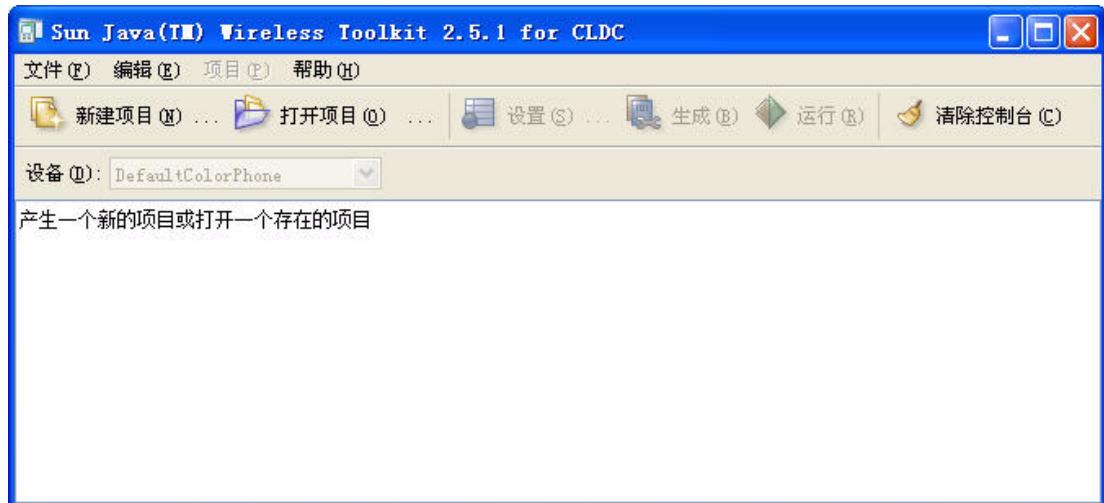
| 演示程序 | API | 描述 | 特别说明 |
|----------------|-----------------|-----------------------------|------|
| SVGContactList | JSR 226 SVG API | 演示以不同样机显示的联系人列表。 | A.20 |
| SVGDemo | JSR 226 SVG API | 展示用于渲染 SVG 的各种技术。 | A.21 |
| UIDemo | MIDP 2.0 | 展现 MIDP 2.0 宏广的用户界面功能 | |
| WMADemo | WMA 2.0 | 显示如何发送和接收 SMS、CBS 和 MMS 消息。 | A.22 |

A.2 一般说明

可以运行并启动大多数演示应用程序，无需进行特别准备。但是，某些演示程序需要更改工具包首选项或设置。本节介绍一般步骤。

第一步是运行工具包。要执行此操作，请转至 Microsoft Windows 的“开始”菜单，然后选择“开始”>“所有程序”>“Sun Java Wireless Toolkit 2.5.2 for CLDC”>“Wireless Toolkit 2.5.2”。将显示用户界面，如图 A-1 中所示。

图 A-1 Wireless Toolkit 用户界面



单击“打开项目”按钮打开演示应用程序。将显示所有可用应用程序的列表。如第 1.1.2 节，第 1-2 页上的“工作目录文件”中所述，安装目录中的项目以斜体显示，而工作目录中的项目以粗体显示（图 1-1）。在安装目录中打开某项目时，将会在工作目录中创建一个副本，然后会打开该副本。

打开应用程序后，可以按工具栏中的“运行”按钮，或者如果需要安装在仿真器上，则单击“项目”>“通过 OTA 运行”

将打开设备仿真器窗口，显示运行中的演示应用程序。如果有 MIDlet 的菜单，请使用导航箭头选择一项，然后选择“选择”按钮。当演示程序运行时，用户可能需要按屏幕下方左侧或右侧的某个软键。用户可使用软键安装或启动应用程序、打开菜单、退出或执行某些其他操作。有一些示例包含这些说明。

某些演示程序需要特定的设置和说明。例如，如果某个演示程序使用 Web 服务并且用户位于防火墙之后，则必须配置仿真器的代理服务器设置，否则演示将失败：

- 选择“编辑”>“首选项”
- 选择“网络配置”图标。
- 选中“使用代理服务器”。
- 填写代理服务器地址字段和端口号。

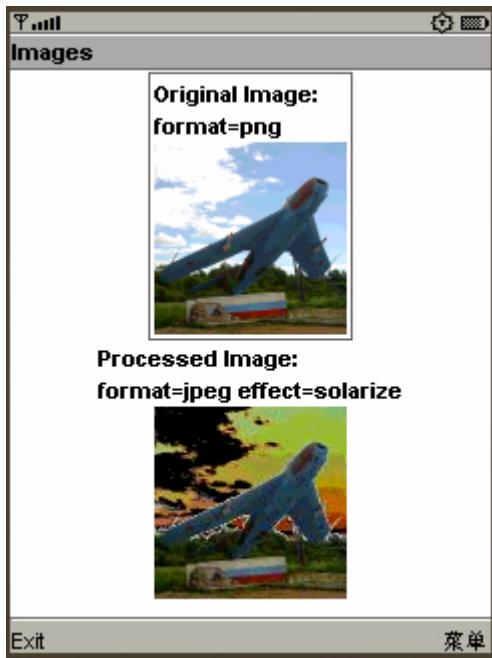
有关详细的操作说明，请阅读每个演示程序的说明。

A.3 Advanced Multimedia Supplements

该 MIDlet 套件演示了 JSR 234 Advanced Multimedia Supplements (AMMS) 的功能。其包含以下 MIDlet：

- **图像效果**——展示标准图像处理操作。从菜单中选择一个效果。将在源图像之后展示处理过的图像。请注意，某些项（例如“设置变换”）可以执行多项操作。单击“完成”软按钮应用每个效果。

图 A-2 处理 MIDlet 中的图像



- 收音机调谐器——通过工具包的内置 HTTP 服务器读取项目目录中的音频文件来模拟收音机调谐器。
- 相机——展示 Advanced Multimedia Supplements 如何控制设备的相机。屏幕将显示相机的取景器（通过视频模拟）。用户可使用菜单中的命令更改相机设置，以及拍摄并管理快照。
- 移动直升机——模拟绕固定观测者飞行的直升机。使用耳机可获得最佳效果。用户可控制许多模拟参数，包括直升机的大小、是否使用多普勒效果以及观测者的方位（例如站直或趴下）。
- 音效——展示 Advanced Multimedia Supplements 的高级音频功能。随着音频文件的不断循环，用户可调整音量、声像、均衡器设置、混响以及合唱设置。

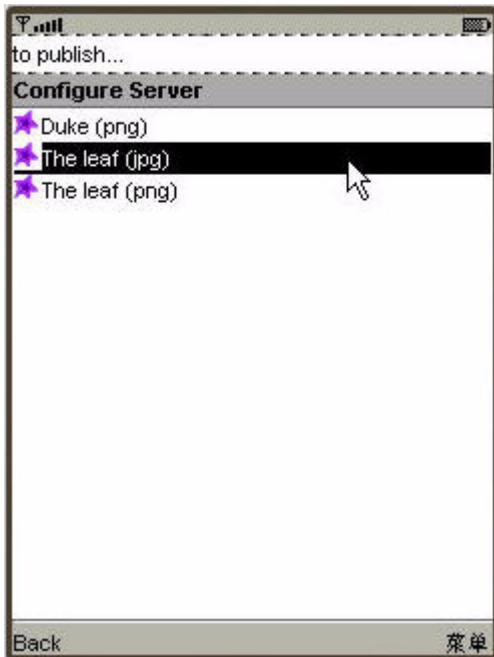
A.4 Bluetooth Demo

该应用程序包含演示如何使用 JSR 82 的蓝牙 API 的 MIDlet。

BluetoothDemo 项目展示如何在两个使用蓝牙的设备之间传输图像。必须运行仿真器的两个实例才能了解该演示程序是如何工作的。

在第一个仿真器中，启动 Bluetooth Demo，然后选择“服务器”。仿真器询问您是否允许进行蓝牙连接时，选择“Yes”。服务器将启动并显示图像列表。起初，蓝牙网络中没有任何图像。要使图像可用，请选择这些图像并在菜单中选择“发布图像”（或键入或单击 1）。图标颜色将从紫色变为绿色，指示图像已发布。

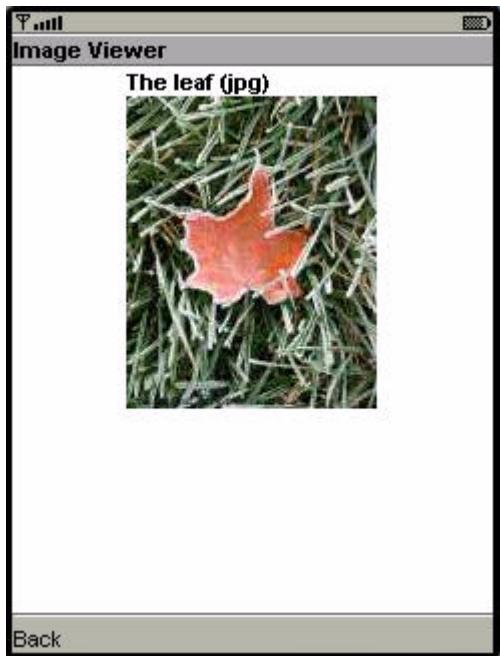
图 A-3 运行 Bluetooth Demo 服务器



在第二个仿真器上，启动 Bluetooth Demo，然后选择“客户机”。MIDlet 会提示您，它已就绪，可进行图像搜索。单击“查找”软按钮。MIDlet 将查找另一个仿真器并从中获取图像列表。从列表中选择一个图像，然后选择“装入”。仿真器询问您是否允许连接时，选择“Yes”。

- 如果在受信任的保护域中运行演示程序，则会使用模拟蓝牙传输图像并在客户机仿真器上显示图像。
- 如果不在受信任的保护域中运行，则第一个仿真器（服务器）将显示提示，询问用户是否授权来自客户机的连接。选择“Yes”。图像将显示在客户机仿真器中。

图 A-4 使用模拟蓝牙传输的图像



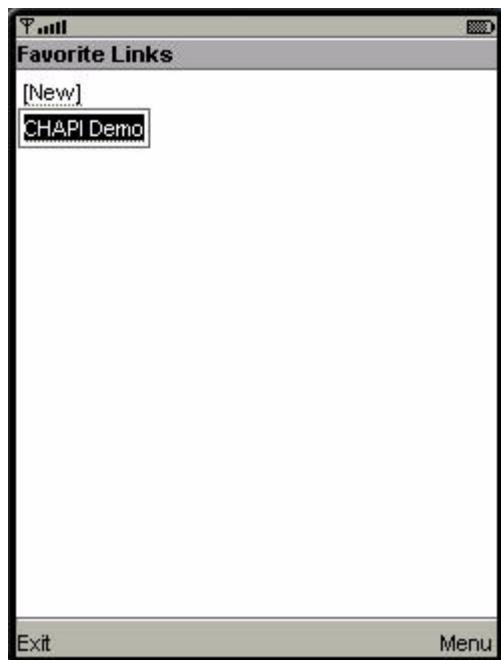
A.5 CHAPIDemo

CHAPIDemo 是内容浏览器（请参见 JSR 211）。它包含一个收藏夹列表，使用户可选择和查看各种内容。

该演示程序使用内容处理器注册表，因此，使用“运行”按钮时，您无法看到其所有功能。而使用“项目”>“通过 OTA 运行”可将应用程序安装到仿真器中。如果要了解具体方法，请阅读[第 2.3.2 节](#)，[第 2-8 页上的“安装”](#)中的介绍。

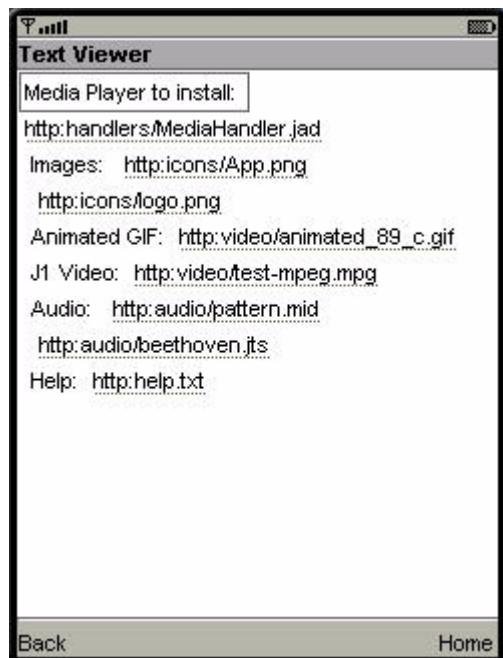
安装 CHAPIDemo 后，它将作为文本查看器显示在应用程序列表中。CHAPIDemo 是 MIDlet，用于纯文本的内容处理器。选择文本查看器，然后从软按钮菜单中选择“启动”。将显示收藏链接列表。

图 A-5 在 CHAPIDemo 中查看收藏链接



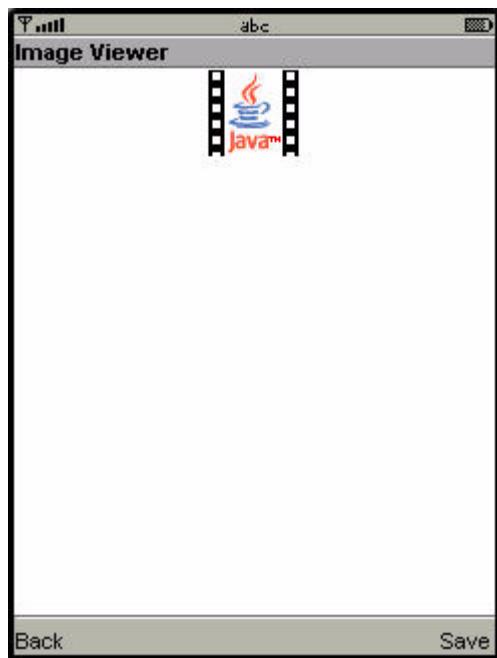
使用导航键高亮显示 CHAPIDemo，然后按仿真器上的“选择”按钮。应用程序将询问您是否可以使用通话时间。按“是”软按钮。将显示各种内容的列表（图 A-6）。

图 A-6 内容列表



请尝试选择一个 Duke.png。使用箭头高亮显示链接，然后按“选择”按钮查看该文件。使用 CHAPI 时，将启动 ImageViewer MIDlet 并显示内容。

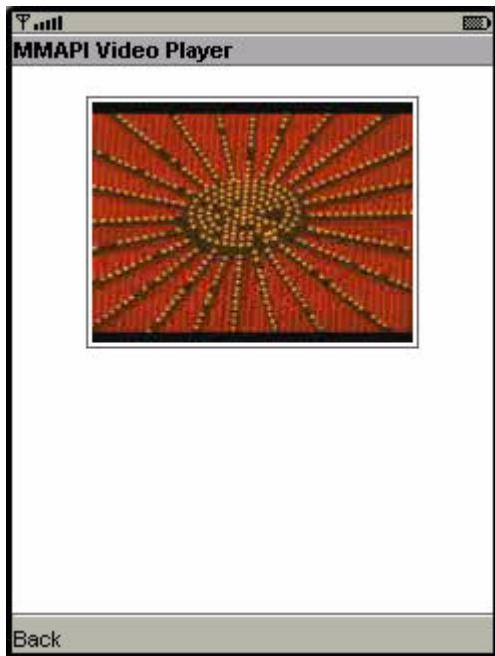
图 A-7 查看 PNG 图像



其他类型的内容需要另一内容处理器 MIDlet 套件 MediaHandler。要从 CHAPIDemo 安装该套件，请选择 `MediaHandler.jad` 链接（图 A-6 中显示的列表中的第一项）。将调用 AMS 并引导您完成安装过程。

安装 MIDlet 套件后，用户即可查看文本查看器中列出的其他类型的内容。例如，选择 `http://video/test-mpeg.mpg` 可查看一系列图像，包括图 A-8 中显示的图像。

图 A-8 使用 MediaHandler 查看 MPEG 视频



要查看 TextViewer 和 ImageViewer MIDlet 的内容处理器设置，请单击“设置”，然后单击“内容处理器”图标。用户可能还要检查 MediaHandler 项目。

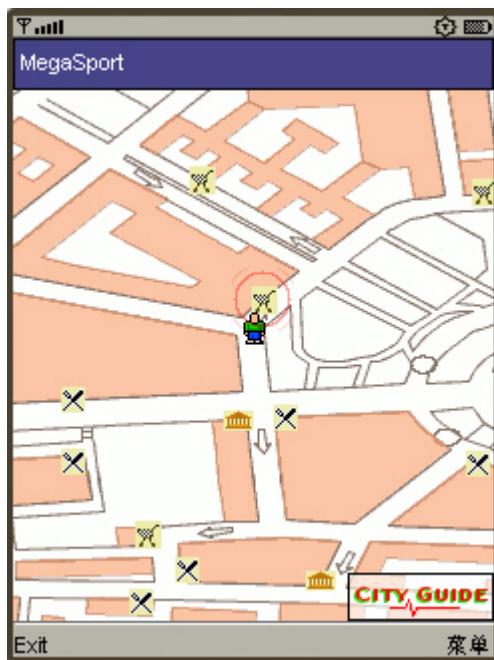
A.6 CityGuide

CityGuide 演示如何使用 Location API (JSR 179)。它将在城市地图上显示步行者当前的位置。步行者会在城市内移动，当步行者接近路标时，路标也会高亮显示并标识。在此演示中，我们可以通过名为 `citywalk.xml`（事件文件）的 XML 脚本（该脚本用于提交设备位置信息）获取步行者的位置。有关完整说明，请参见第 13 章。

由于时常进行位置提示，因此最好在制造商（受信任）模式下运行该演示程序，如第 6.2.1 节，第 6-3 页上的“MSA 保护域”中所述。在用户界面中，选择“编辑”>“首选项”，然后选择“安全”。选择安全域的制造商。

打开并运行 CityGuide 项目。在仿真器中，启动 CityGuide MIDlet。单击“下一步”查看地图页。

图 A-9 您在城市中的位置



从仿真器窗口菜单中选择 "MIDlet" > “外部事件”。在“位置”选项卡上，单击浏览按钮。选择 `workdir\apps\CityGuide\citywalk.xml` 中的以下事件文件。

现在，位于窗口底部的播放器按钮将处于活动状态。请参见图 13-1。按绿色播放按钮（向右指的三角形）运行脚本。

显示屏将显示四种路标：餐馆、博物馆、商店和电影院。要调整路标显示，请打开软菜单，然后选择“设置”命令。请参见图 A-10。使用导航键高亮显示某一类别，然后使用“选择”按钮选中或取消选中某一项。

当用户位于路标（在地图中高亮显示）附近时，可打开软菜单并选择“详细信息”命令以查看更多信息。有关位置脚本的详细信息，请参见第 13 章。

图 A-10 位置设置

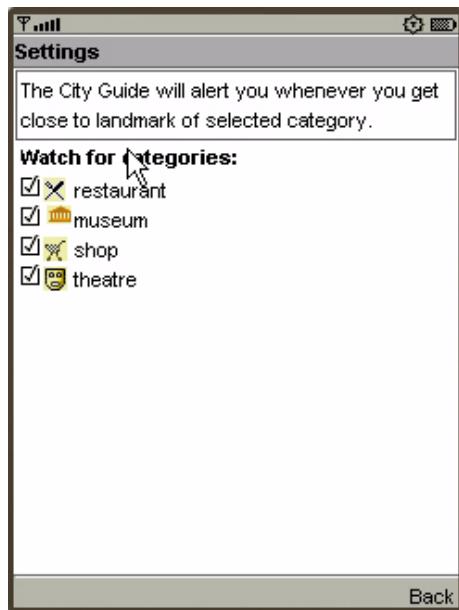
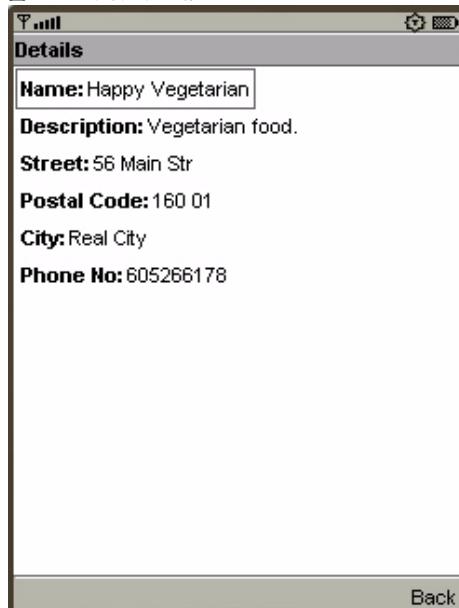


图 A-11 路标详细信息



A.7 演示

该演示包含重点说明不同 MIDP 功能的几个 MIDlet。

A.7.1 颜色

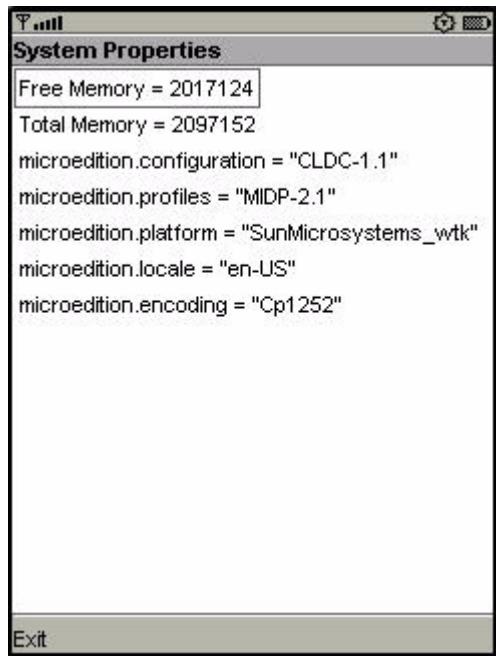
该应用程序会显示一个较大的横向矩形，跨越整个屏幕宽度。下面有十个较小的纵向矩形，跨越整个屏幕。最下面是三个横向的颜色栏，分别表示蓝、绿和红 (RGB) 的值。根据第一个菜单选择，这些值以十进制 (0-255) 或十六进制 (00-ff) 来表示。

- 要选择纵向栏以进行更改，请使用向上导航箭头移动到颜色栏。使用向右导航箭头高亮显示颜色栏。较大的矩形将变为所选颜色栏的颜色。
- 使用上或下选择箭头选择要更改的值（红、绿或蓝）。使用左箭头键或右箭头键增加或减小所选值。第二个菜单项允许用户以 4（微调）或 32（粗调）的增量进行增减。
- 用户可更改任何纵向栏的颜色。

A.7.2 属性

该 MIDlet 显示属性值。例如，请参见图 A-12：

图 A-12 系统特性



A.7.3 Http

该测试应用程序使用 HTTP 连接请求 Web 页面。该请求是通过 HTTP 协议 GET 或 POST 方法发布的。如果使用 HEAD 方法，则将从请求读取头特性。

准备运行演示

开始前，请根据如下所述检查设置：

- 转至“首选项”>“安全”。将策略设置为 JTWI，并将域设置为最大值。
- 在“首选项”>“网络配置”中，HTTP 的版本必须为 1.1。
- 如果位于防火墙之后，请转至“首选项”>“网络配置”，然后指定代理服务器信息。
- 如果运行了防病毒软件，则可能需要创建一种规则，使该 MIDlet 允许连接到特定 Web 站点或允许来自特定 Web 站点的连接。

运行演示

启动 **Http** MIDlet。默认情况下，MIDlet 将尝试联系 `http://www.yahoo.com`。要进行测试，请选择“菜单”软键，然后选择 1、2 或 3 以测试选定的 URL。

Http 测试将返回其可获取的信息。如果屏幕中显示这些信息，请使用下箭头滚动到最后。信息量取决于请求的类型以及该页所提供的 META 信息量。要提供主体信息或内容，该页必须根据 RFC 2616 中所述声明 `CONTENT-LENGTH`。

使用菜单选项

使用“菜单”软键进行以下操作。

- 选择 1 从所选页面中获取 GET 信息。
- 选择 2 从所选页面中获取 POST 信息。
- 选择 3 显示页面的 HEAD 属性。
- 选择 4 显示 Web 页面的当前列表。用户可选择新页面或将自己的页面添加到列表。要指定新 URL，请选择“菜单”软键，然后选择 4。屏幕将显示 `http://`。键入 URL 剩余部分，并确保以斜线 (/) 结尾。例如 `http://www.internetnews.com/`。按“确定”软按钮。**Http** 测试屏幕将显示新 URL 并提示用户进行操作。

A.7.4 FontTestlet

该 MIDlet 显示各种可用字体：比例、常规、常规斜体、粗体常规和粗斜体。从菜单中选择 1 或 2 以在系统字体 (sans serif) 和等宽字体之间进行切换。

A.7.5 股票行情

与 **Http** 演示程序类似，该样例使用 HTTP 连接来获取信息。使用第 A.7.3 节，第 A-15 页上的“**Http**”中所述的准备步骤。

运行演示项目并启动 Stock MIDlet。

默认情况下，屏幕顶部将显示空的自动收录器栏。在自动收录器下方，菜单列表显示四个应用程序：**Stock Tracker**、**What If? Alerts** 和 **Settings**。在可以使用前三个应用程序前，必须添加股票名称。

A.7.5.1 使用设置

要使用应用程序功能，必须提供一些股票名称以供应用程序进行操作。

向自动收录器添加股票名称

要向自动收录器添加股票名称，请使用导航箭头选择“设置”。

选择“添加股票”。

显示屏中将提示用户输入股票名称。键入 SUNW 并选择“完成”软键。现在，所增加的股票及其当前价格将显示在自动收录器中。用户可再多添加一些股票名称，例如 IBM 和 HPQ。

更改更新时间间隔

默认情况下，更新时间间隔为 15 分钟。选择“更新”更改时间间隔。使用导航箭头选择以下选项之一：连续、15 分钟、30 分钟、1 小时或 3 小时。选择“完成”软键。

删除股票

选择“删除股票”。可以看到已添加的股票列表。使用导航键选择一个或多个要删除的股票。选择“完成”软键。

A.7.5.2 股票跟踪器

股票跟踪器显示已添加股票的列表及这些股票的当前价格。股票跟踪器显示有关所选股票的其他信息，例如，上次交易情况及较高价格和较低价格。

选择股票，然后按“选择”按钮。

A.7.5.3 What If?

What If? 是一种应用程序，会询问用户所持有股份的原始购买价格和股份数目。他会根据当前价格计算您的盈利或亏损。

选择股票名称。

输入购买价格和股份数目，然后按“计算”。

A.7.5.4 Alerts

该应用程序在价格更改为用户指定的值时会向用户发送通知。

从主菜单中，选择“警报”。

选择“添加”。

选择股票。屏幕将提示 Alert me when a stock reaches。输入整数。

警报置于“当前警报”列表中。要删除警报，请按“删除”，然后选择警报。选择“完成”软键。

当达到指定的价格时，用户将听到铃声并收到一条消息。例如，所选股票名称已达到价格点 `$value`，并且当前交易价格为 `$current_value`。触发警报后，该警报将从“当前警报”列表中消失。

A.7.6 证明书

这演示了在线证明书拍卖应用程序的执行方式。主屏幕顶部将显示证明书自动收录器。默认情况下，“选择乐队”字段显示 "Alanis Morissette"。

要选择乐队，请高亮显示乐队名称，然后按“选择”按钮。使用下箭头高亮显示其他乐队（例如 `moby`），然后按“选择”按钮。将显示可用拍卖。

要进行竞标，请选择“菜单”软键，然后选择 2。使用箭头键逐字段进行移动。填写每个字段。选择“下一步”软键。应用程序将要求用户确认竞标。使用箭头键高亮显示“提交”，然后按“选择”按钮。用户将收到确认号。单击“乐队”返回欢迎页面。

要设置警报，请选择“菜单”软键，然后选择 3。使用导航箭头移动到字段，然后键入一个比当前竞标报价高的价格。选择“保存”软键。用户将返回欢迎页面。只要以超出警报值的价格进行竞标，就会触发警报。用户的设置决定应用程序检查更改的频率，因此可能要过几分钟才能听到警报。

要添加乐队，请选择“菜单”软键，然后选择 4。键入乐队名称或逗号分隔的名称列表。选择“保存”软键。确认后，将返回欢迎页面。所添加的乐队将显示在“选择乐队”下拉列表中。

注 - 这只是一个演示程序。要充分地描述乐队，必须编辑文件 `workdir\apps\Demos\src\example\auction\NewTicketAuction.java`。

要删除乐队，请选择“菜单”软键，然后选择 5。导航到某个乐队，然后选择“选择”按钮标记复选框。用户可选择多个乐队。选择“保存”软键。

要显示自动收录器显示屏、更新、警报音量和日期的当前设置，请选择“菜单”软键，然后选择 6。如果需要，可使用箭头键和选择键更改这些值。选择“保存”软键。

A.7.7 ManyBalls

该 MIDlet 一开始是一个球在屏幕中滚动。使用上箭头和下箭头提高或降低球速 (fps)。使用右箭头或左箭头增大或减少球数。

A.8 Demo3D

该应用程序包含三个 MIDlet，这三个 MIDlet 展示了仿真器对 Mobile 3D Graphics API (JSR 184) 的支持。

A.8.1 Life3D

Life3D 实现了流行的三维“生命游戏”。一个立方块代表一个活细胞。每个细胞周围最多有 26 个细胞（包括对角线）。在动画的每个步骤中，如果一个活细胞周围的细胞少于 4 个，该细胞会由于孤独而死去，但若是超过了 5 个，则又会由于过度拥挤而死去。一个空细胞周围恰好有 4 个细胞时，则会变成一个新的活细胞。

游戏板的查看角度缓慢旋转，让您可以从各个角度查看游戏板。

图 A-13 三维“生命游戏”

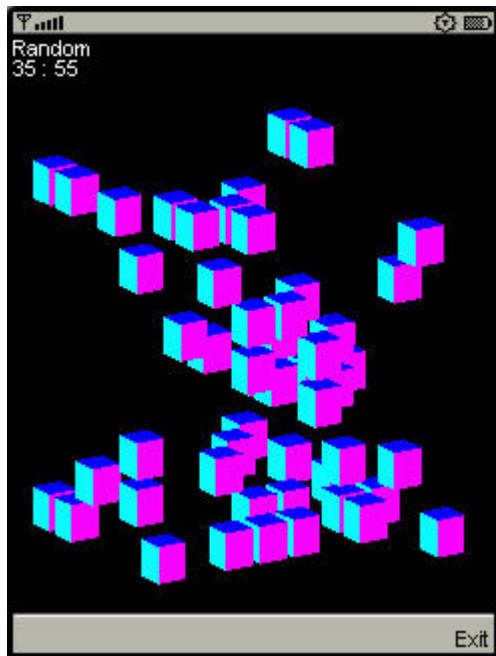


表 A-2 中小键盘上的按钮可控制游戏。

表 A-2 控制 Life3D

| 按钮 | 描述 |
|----|---|
| 0 | 暂停动画。 |
| 1 | 恢复动画的默认速度。 |
| 2 | 加快动画速度。 |
| 3 | 减慢动画速度。 |
| 4 | 从任意列表中选择上一预设配置。配置的名称显示在屏幕顶部。 |
| 5 | 从列表中选择下一预设配置。 |
| * | 随机生成一种配置并动画显示，直到其稳定或者死亡。如果它死亡，则随机生成一种新配置。 |

该示例的源代码有相当完整的文档记录。请参见 *toolkit\apps\Demo3D\src\com\superscape\m3g\wtksamples\life3d\Life3D.java*。

A.8.2 PogoRoo

PogoRoo 向您展示一只袋鼠在娃娃跳上上下弹跳。要控制这只袋鼠，请使用箭头键。按上箭头键可以向前移动，按下箭头键可以向后移动，按左箭头键和右箭头键可以改变方向。用户可能需要按住键来观看效果。

图 A-14 弹跳的袋鼠



A.8.3 retainedmode

`retainedmode` MIDlet 会播放一个场景文件，内容显示的是一个不知疲倦的滑板手在无尽的环道中不停地滑行。

图 A-15 不知疲倦的滑板手



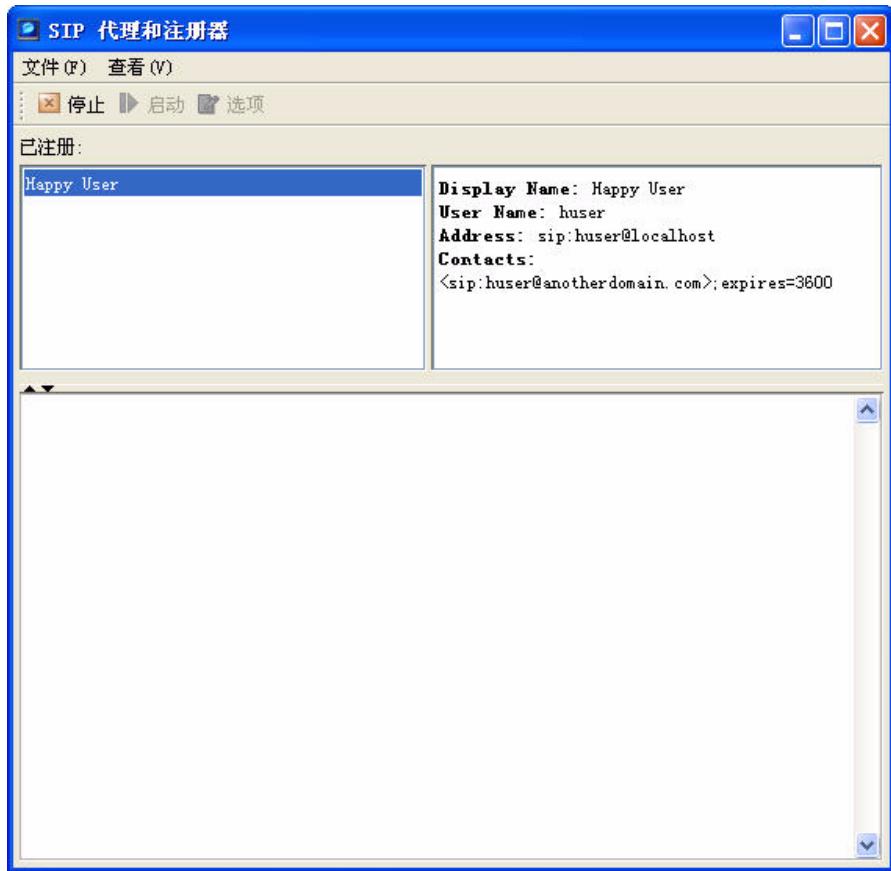
A.9 GoSIP

GoSIP 是一种聊天应用程序，它使用 SIP (JSR 180) 设置使用 SIP 代理服务器和注册器的通讯。

通过运行 SIP 服务器开始。选择“文件”>“实用程序”。选择“启动 SIP 服务器”，然后按“启动”。将显示 SIP 代理服务器窗口。单击“开始”运行服务器。

接下来，通过 GoSIP 应用程序运行仿真器的两个实例。

图 A-16 SIP 代理服务器和注册器



在第一个仿真器中，启动 Sippy A。当系统提示输入代理主机时，输入本地计算机的名称或 IP 地址并选择“下一步”，然后选择“注册”。在 SIP 服务器窗口中，将显示来自仿真器的 SIP 消息。Sippy A 显示在已注册用户的列表中。仿真器将建议用户邀请他的朋友 Sippy B 进行聊天。暂且不要执行此操作。

在第二个仿真器中，启动 Sippy B。与前面的一样，输入 SIP 代理的地址，选择“下一步”，然后选择“注册”。SIP 服务器窗口中将显示 Sippy B 用户。

在第一个仿真器中，选择“邀请”。第二个仿真器将指示它正响铃。选择“应答”开始聊天。两个仿真器现在均显示聊天屏幕。用户可使用“发送”命令来回发送消息。

完成后，选择“再见”结束聊天。

A.10 i18nDemo

该 MIDlet 套件展示了 JSR 238 Mobile Internationalization API。该 MIDlets String Comparator 和 Formatter 显示如何针对不同的语言环境对字符串和显示号码进行相应地排序。第三个 MIDlet (MicroLexicon) 是一个小型的短语翻译工具，如果在布拉格、海尔兹利亚、北京、米兰或其他一些地方，您需要点一杯啤酒时，它可以派上用场。

注 - Sun Java™ Wireless Toolkit for CLDC 的默认字体不支持中文和日文。要使用这些语言，请在运行该演示之前按照以下步骤进行操作：

1. 安装支持中文或日文的 True Type 字体。
 2. 修改 `toolkit\wtklib\devices\skin-directory\skin.properties` 以指定该字体。
-

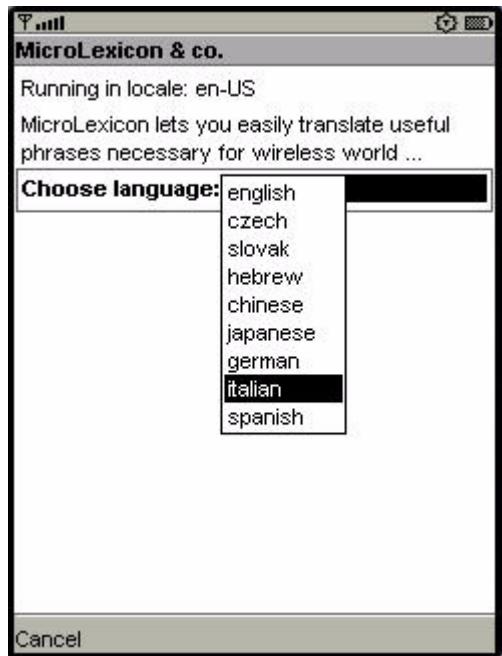
要运行 MIDlet，请使用“选择”按钮高亮显示 MIDlet，然后使用右下方的按钮启动 MIDlet。

String Comparator MIDlet 演示了字符串（城市名称）的排序如何因语言环境而有所不同。启动 MIDlet。使用右下方的按钮查看菜单。单击或键入 2 选择“排序 - 默认”，然后按字母顺序对列表进行排序。单击或键入 3 选择“排序 - 斯洛伐克语”。无论顶部有没有标记，都可以轻松看出以字母 Z 开头的城市之间的区别。单击“退出”返回 MIDlet 列表。

第二个 MIDlet (Formatter) 只是针对不同语言环境的格式显示时间和数字。单击“下一步”以查看所有四个屏幕。单击“退出”返回 MIDlet 列表。

最后一个 MIDlet (MicroLexicon) 将短语从一种语言翻译成另一种语言。要从列表中选择目标语言，请使用导航箭头高亮显示“选择语言”。单击“选择”按钮查看语言下拉列表。使用导航箭头选择语言（请参见图 A-17），然后单击“选择”按钮。

图 A-17 选择目标语言



MicroLexicon 显示一列短语。高亮显示一个短语，然后按仿真器上的“选择”按钮。用户可以看到目标语言的标志以及翻译的短语。

要更改源语言，请选择“编辑”>“首选项”。单击“国际化”选项卡，然后输入有效的语言环境字符串。下次运行仿真器和 MicroLexicon 时，如果支持，说明文字将以给定的语言环境显示。例如，cs-CZ 就可行。

MicroLexicon 由 MIDlet 资源支持。要了解如何使用工具包对应用程序进行本地化，请选择“项目”>“国际化资源管理器”。由 MicroLexicon 使用的所有资源（包括文本和图像）均会显示。用户可以编辑资源并再次运行 MicroLexicon，看看有什么变化。由于资源是在运行时装入的，因此无需再次生成应用程序。

图 A-18 国际化资源管理器



这些资源本身存储在 `workdir\apps\i18nDemo\res\global` 中。

A.11 JBricks

JBricks 是一种游戏，演示如何使用 JSR 229 Payment API。游戏本身类似于 Breakout 或 Arkanoid。在 JBricks 中，用户可购买其他生命或新的游戏级别。Payment API 将在后台处理这些细节。

要使用 JBricks 的付费功能，请使用“项目”>“通过 OTA 运行”将 JBricks 安装到仿真器中。如果要了解具体方法，请阅读[第 2.3.2 节，第 2-8 页上的“安装”](#)中的介绍。

要查看 JBricks 如何使用 Payment API，请从游戏的主菜单中选择“Buy Life”或“购买级别”。接下来，选择是购买一条生命还是以优惠价格一次购买三条生命。在下一屏幕上用户可选择付费类型。

图 A-19 选择付费类型



使用导航箭头可以选择以“付费方式”开头的行。单击“选择”按钮查看下拉菜单中可能的信用卡适配器。使用导航箭头选择“VISA 适配器”，然后单击“选择”按钮。单击右下方的“是”以继续。

接下来，用户可以输入信用卡信息。使用任何有效的 VISA 号码（例如 4111111111111111）和有效的截止日期。

图 A-20 提供付费信息



要查看仿真器当前实例的事务，请选择 "MIDlet" > “外部事件”，然后单击“付费事务”选项卡。将显示该仿真器特定实例的事务。

图 A-21 查看事务



另外，用户可以查看通过工具包付费系统的所有事务。选择“文件”>“实用程序”，然后选择“付费控制台”。控制台中的事务如下：

```
PSP 控制台正在运行，使用电话号码 +5550001。
PSP 服务器正在 https://localhost:-1 上运行
收到来自 127.0.0.1 的付费请求
    信用卡发行者： VISA
    信用卡类型： 0
    信用卡号： 4111111111111111
    信用卡持卡人： Jonathan Knudsen
    功能 ID: 3_lives
    信用卡验证号 (CCV)： 123
    有效负载： null
响应 127.0.0.1
HTTP/1.1 200 就绪
内容长度： 0
付费响应： 成功
付费时间戳： 1156282954734
```

A.12 JSR172Demo

JSR172Demo 展示如何通过 MIDlet 访问 Web 服务。Web 服务已经运行在某个 Internet 服务器上。如果位于防火墙之后，则必须配置仿真器的代理服务器设置。选择“编辑”>“首选项”，然后选择“网络配置”。填写代理服务器地址文件和端口号。生成并运行该示例。

JSR172Demo 包含一个名为 Server Script 的 MIDlet。请启动它并按照提示进行操作。用户可浏览模拟的新闻头条，这些新闻头条都是从 Web 服务中检索出来的。

要查看详细的内幕，请使用网络监视器。

A.13 MobileMediaAPI

MobileMediaAPI 应用程序包含四个展示工具包多媒体功能的 MIDlet。本节介绍 MIDlet，同时还包括有关如何在应用程序中使用多媒体的其他信息。

A.13.1 Simple Tones

Simple Tones 示例演示了如何使用交互合成铃声。选择一个示例，然后单击右下方的“播放”。

- Short Single Tone 和 Long Single Tone 使用 Manager.playTone() 来播放具有不同音高的铃声。
- 短 MIDI 事件在交互式 MIDI 设备（定位器 "device://midi"）上播放和弦。MIDIControl 的 shortMidiEvent() 方法用来触发和弦音符。
- 要运行 MMAPI Drummer 演示，请单击或键入数字键 (0-9)。每个数字播放一个音。

A.13.2 Simple Player

Simple Player 应用程序演示了仿真器音频和视频的功能范围。它包括各种格式的样例文件，并能播放仿真器永久存储或 HTTP URL 上的文件。

播放器部分使用通用的 `javax.microedition.media.Player` 接口。播放器显示持续时间、媒体时间以及用于运行媒体文件的控件。如果文件中有元数据，则可以在播放器中查看信息，如作者和标题。在 MIDI 文件中，如果文件中出现卡拉OK文本，则播放时这些文本就会显示在屏幕上。如果适用，可以在显示屏上看到图形用户界面控件。用户可通过以下方式来访问这些控件：在 Simple Player 中选择一个媒体样例，然后按“菜单”按钮查看并选择所需的命令。

选择 Simple Player，然后单击“启动”。此演示包括以下媒体样例：

- **Bong** 播放短 WAV 文件。可以对某些回放功能进行调整，本文后面的内容对此进行了说明。显示屏使用分 : 秒 :1/10 秒的格式来显示声音的持续时间，例如 00:17:5。该音频样例是 MIDlet 套件 JAR 文件中的一个资源文件。
- **MIDI Scale** 播放音阶样例。显示屏显示所选音乐文件的标题、歌曲的持续时间、已回放时间以及当前节奏（每分钟节拍数，bpm）。该 MIDI 文件存储在 MIDlet 套件 JAR 文件中。
- **Simple Ring Tone** 播放一小段贝多芬的第五交响曲。显示屏显示所选音乐文件的标题、歌曲的持续时间、已回放时间（以秒和 1/10 秒表示）以及当前节奏（每分钟节拍数，bpm）。该铃声文件 (.jts 格式) 存储在 MIDlet 套件 JAR 文件中。
- **WAV Music** 播放一小段音频文件。显示屏显示该音频文件的标题、音频持续时间、已回放时间以及回放速率（以百分比表示）。可从 HTTP 服务器检索到该 WAV 文件。
- **MIDI Scale** 播放从 HTTP 服务器检索到的 MIDI 文件。
- **Animated GIF** 示例展示了一个从 1 数到 5 的 GIF 动画，该文件存储在 MIDlet 套件 JAR 中。
- **Audio Capture** 是默认设备的功能，您可以利用它从话筒或者连接的设备捕获音频。声音被捕获的同时从扬声器中播放出来。为了避免回音，请使用耳机或听筒。
- **Video Capture Simulation** 模拟如何查看输入视频，例如在装备了相机的设备上查看。
- **MPEG1 Video [http]**。播放在 <http://java.sun.com/products/java-media/mma/media/test-mpeg.mpg> 上找到的 MPEG 视频。
- **[enter URL]** 允许从任意 HTTP 服务器播放媒体文件。在插入点处键入有效的 URL（例如 <http://java.sun.com/products/java-media/mma/media/test-wav.mpg>），然后单击“确定”播放文件。如果要打开某个 HTTP 目录以从中选择媒体，请确保在 URL 的末尾加上一条斜线。

另外，Simple Player 用铃声文本传输语言 (RTTTL) 来解析铃声。有关 RTTTL 的信息，请参见 <http://www.convertyourtone.com/rtttl.html>。

Simple Player 包括通用命令集，可以控制媒体回放。可以从 Simple Player 菜单使用这些命令，有些命令还具有相关联的小键盘按钮。下表对这些命令进行了说明。

表 A-3 Simple Player 命令

| 命令 | 键 | 描述 |
|-----------|-----------|---|
| 静音 / 取消静音 | 0 | 关闭声音，但继续播放文件。此命令可切换到“取消静音”。 |
| 音量 | * 和 # | 升高或降低响度。 |
| 元数据 | | 显示媒体文件提供的信息，例如，版权信息、标题以及音轨列表。 |
| 5 秒钟后停止 | | 在回放过程中设置此项后，将在 5 秒钟后暂停音频播放。 |
| 速率 | 4 和 6 | 改变回放的速率。 |
| 节奏 | | 提高或降低铃声序列或 MIDI 文件的节奏。 |
| 音高 | 向上和 向下 | 降低或升高 MIDI 文件中的音符。 |
| 开始 / 停止录音 | | 对音频回放进行录音。创建一个包含所录音频的文件，该文件位于仿真器运行的目录下。如果不指定文件名，则文件将被命名为 recording.wav。此命令可切换到“停止录音”。 |
| 单格放映 | 7 和 9 | 在视频文件中，一次向前或者向后跳一帧。 |
| 播放 / 停止 | 2 和选择 | 开始或停止媒体播放。 |
| 循环模式 | | 播放完成后，立即回放音频文件。运行一次“循环模式”，播放一次音频文件。按第二次将播放文件三次。按第三次将重复播放文件。按第四次将返回到单一播放模式。 |
| 跳帧 | 1 和 3 | 向前或向后跳跃整个媒体播放进度的 5%。声轨与视频同步。 |
| 重绕 | | 返回到音频回放的开始时间。 |
| 停止并重绕 | 5 | 停止回放并重绕到开始时间。 |
| 快速帮助 | | 显示命令和小键盘按钮列表。 |

这些命令适用与否取决于 Simple Player 正在播放的媒体类型。此外，一些命令可以通过使用小键盘按钮来调用。下表介绍了命令的可用性、命令对应的小键盘按钮以及 MMAPI 中相关的类。

请注意，Simple Player 应用程序本身会提供包含命令及其相应小键盘按钮的短列表。只需从菜单中选择“快速帮助”命令即可。

A.13.3 Video

Video 应用程序演示了仿真器具备了怎样的播放 GIF 动画文件和捕获视频的功能。可以在装备了相机的真实设备上使用视频捕获将相机所拍摄的内容展现给用户。

GIF 动画和视频捕获可以通过 Form Item 或 Canvas 来实现。Video 演示程序包括所有可能性。Animated GIF - Form [jar] 将 GIF 动画显示为 Form Item。窗体中还包含一些有关回放的信息，包括当前时间。选择“快照”命令对运行中的动画拍摄快照。拍摄的快照将被放在 GIF 动画后面的窗体中。

- Animated GIF - Canvas [jar] 在 Canvas 中播放 GIF 动画。一个简单的指示符说明动画的进度。选择“快照”可以获取当前画面的静态图像。快照短暂显示过后，显示屏又会回到动画。
- Video Capture - Form 模拟如何从相机或其他来源捕获视频，并将其显示为 Form 中的一个 Item。选择“快照”命令对捕获的视频拍摄快照。拍摄的快照将被放在视频捕获后面的窗体中。
- Video Capture - Canvas 模拟如何从相机或其他来源捕获视频，并在 Canvas 中显示。选择“快照”可以获取当前画面的静态图像。快照短暂显示过后，显示屏又会回到视频捕获。
- MPEG1 Video - Form、MPEG1 Video - Canvas

MPEG1 应用程序从 Web 获取 MPEG，因此如果用户位于防火墙之后，则必须配置仿真器的代理服务器设置。

选择“编辑”>“首选项”，然后选择“网络配置”。选中“使用代理服务器”。填写代理服务器地址字段和端口号。对于该演示，请选择“HTTP/1.0”。

当播放演示时，需等待几秒钟，WTK 将获取数据。MPEG1 演示分别与 Video Capture - Form 和 Video Capture - Canvas 具有相同的行为方式。

A.13.4 暂停音频测试

该 MIDlet 用于演示在暂停的 MIDlet 尚未停止其正在运行的 Player 时，Sun Java™ Wireless Toolkit for CLDC 将如何向用户发出警告。启动 MIDlet 后，选择“播放”命令以开始播放某些音频。屏幕显示状态，“正常”或“不正常”。

从仿真器窗口的菜单中选择“MIDlet”>“暂停”。如所预期的，MIDlet 被暂停，并且工具包控制台中不显示任何消息。从仿真器窗口的菜单中选择“MIDlet”>“恢复”。

现在选择“不正常”命令。再次暂停该 MIDlet。在工具包控制台中，用户可看到以下警告：An active media (subtype Player) resource was detected while the MIDlet is paused. Well-behaved MIDlets release their resources in pauseApp(). (在 MIDlet 暂停时检测到活动媒体 [子类型 Player] 资源。正常的 MIDlet 释放在 pauseApp() 中的资源。)

A.13.5 MobileMediaAPI 的属性

MobileMediaAPI 应用程序具有以下属性，用户可在项目设置对话框“用户定义的”选项卡中对这些属性进行修改：

表 A-4 MMAPI 专用的 MIDlet 属性的说明

| 属性 | 描述 |
|------------------------|--|
| PlayerTitle- <i>n</i> | 要通过 Simple Player MIDlet 播放的第 <i>n</i> 个媒体标题的名称。 |
| PlayerURL- <i>n</i> | 要通过 Simple Player MIDlet 播放的第 <i>n</i> 个媒体标题（即 PlayerTitle- <i>n</i> ）的位置。 |
| VideoTest- <i>n</i> | 要通过 Video 应用程序播放的第 <i>n</i> 个媒体标题的名称。 |
| VideoTest-URL <i>n</i> | 要通过 Video 应用程序播放的第 <i>n</i> 个媒体标题（即 VideoTest- <i>n</i> ）的位置。 |

A.14 网络演示

该演示有两个 MIDlet：套接字演示和数据报演示。每个演示需要用户运行两个仿真器实例，以便可以模拟服务器与客户机的关系。

A.14.1 套接字演示

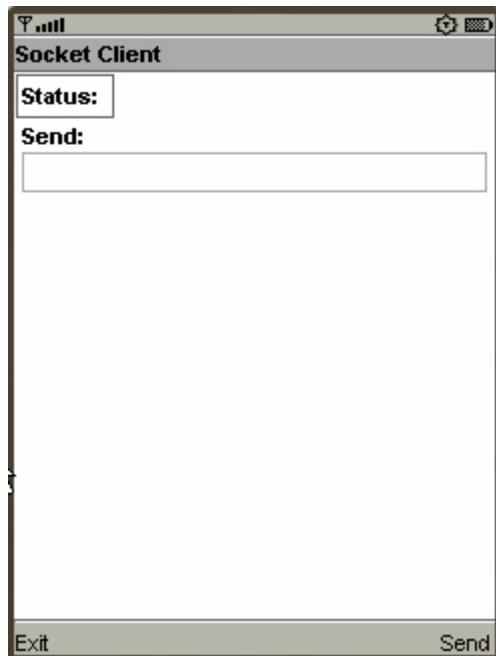
运行仿真器的两个实例。一个作为套接字服务器，另一个作为套接字客户机。

在第一个仿真器中，启动应用程序，然后选择“服务器端”。选择“启动”。仿真器说明演示要通过网络发送和接收数据，并询问“是否可以使用网络？”。选择“Yes”。套接字服务器将显示一个屏幕，表明其正在等待连接。

在第二个仿真器中，启动应用程序，选择“客户机端”，然后选择“启动”。仿真器说明演示要通过网络发送和接收数据，并询问“是否可以使用网络？”。选择“Yes”。套接字客户机将显示一个屏幕，表明其已连接到服务器。使用向下导航箭头可高亮显示“发送”框。在“发送”框中键入消息，然后选择“发送”软键。

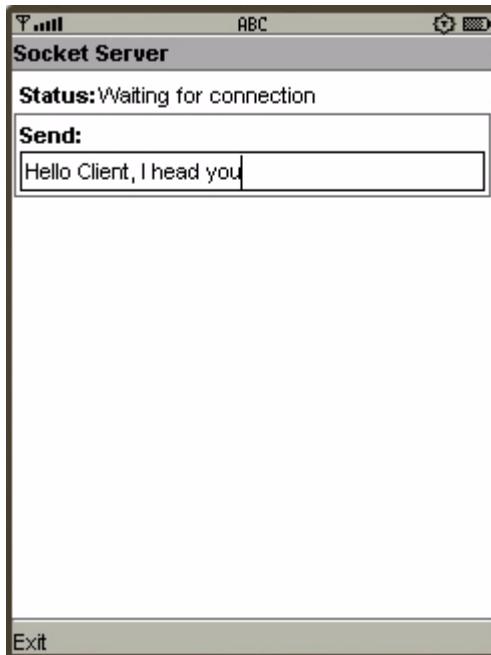
例如，在客户机中，在“Send”框中键入 Hello Server（请参见图 A-22）。选择“Send”软键。仿真器在传送过程中会亮起蓝色灯。

图 A-22 从套接字客户机发送消息



在运行套接字服务器的仿真器上，状态如下：Message received - Hello Server。用户可使用下箭头移动到 "Send" 框，然后键入回复。例如 Hello Client, I heard you。选择 "Send"。请参见图 A-23。

图 A-23 服务器显示收到的消息和要发送的消息



返回套接字客户机，状态将显示从服务器收到的消息。在发送新的消息之前，“Send”框中将保留发送的上一条消息。

A.14.2 数据报演示

该演示类似于套接字演示。

运行仿真器的两个实例。一个作为数据报服务器，另一个作为数据报客户机。

在第一个仿真器中，启动数据报演示，然后选择“服务器端”。选择“启动”。仿真器说明演示要通过网络发送和接收数据，并询问“是否可以使用网络？”。选择“Yes”。最初，数据报服务器的状态为Waiting for connection，并且“发送”框为空。

在第二个仿真器中，启动数据报演示，选择“客户机端”，然后选择“启动”。仿真器说明演示要通过网络发送和接收数据，并询问“是否可以使用网络？”。选择“Yes”。数据报客户机的状态为：Connected to server。使用向下导航箭头可高亮显示“发送”框。在“发送”框中键入消息，然后选择“发送”软键。例如，键入Hello datagram server。

在运行数据报服务器的仿真器上，状态显示如下：Message received - Hello datagram server。用户可使用下箭头移动到“发送”框，然后键入对客户机的回复。

在数据报客户机中，状态字段显示从服务器收到的消息。“发送”框中将保留发送的上一条消息。

A.15 ObexDemo

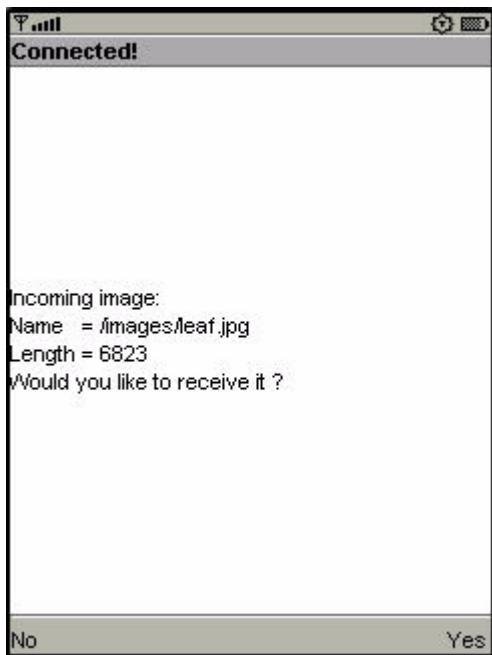
该应用程序展示如何使用 OBEX API 在仿真器实例之传输图像文件。该演示程序展示如何通过模拟的红外连接使用 OBEX。

运行仿真器的两个实例。一个侦听传入连接，另一个尝试发送图像。在第一个仿真器中，启动应用程序，选择“Obex Demo”，然后选择“接收图像”。仿真器说明 OBEX 连接允许其他设备与您的设备进行通信，并询问“是否要进行连接？”。选择“Yes”。侦听器仿真器将显示一个屏幕，表明其正在等待传入连接。

在第二个仿真器（发送方）中，启动 Obex Demo，然后选择“发送图像”。用户将看到一个图像列表。选择一个图像，然后选择“发送”。仿真器说明演示要进行传出客户机连接，并询问是否可以进行连接。选择“Yes”。发送图像实用程序将上载图像。

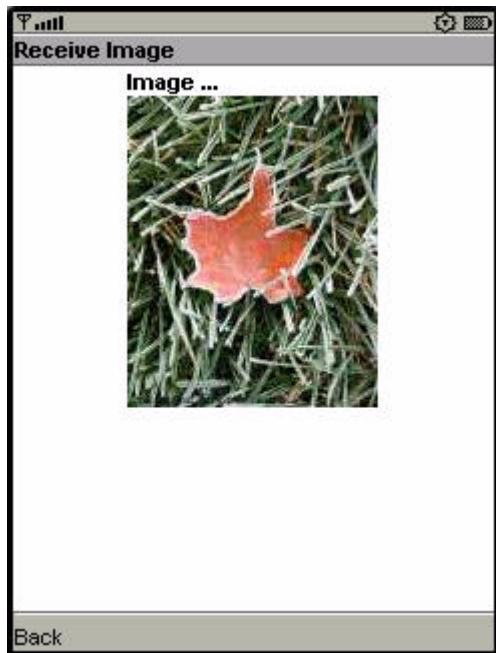
在侦听仿真器中，实用程序将显示有关传入图像的信息，并询问“Would you like to receive it？”请参见图 A-24。

图 A-24 侦听器提示接受连接



选择“Yes”。所选图像通过模拟的红外链接进行传输，并显示在第一个仿真器上。请参见图 A-25。

图 A-25 已成功传输图像



A.16 PDAPDemo

PDAPDemo 展示如何使用 PIM 和 FileConnection API，二者均为 JSR 75 规范的一部分。

A.16.1 浏览文件

要运行文件浏览器，用户需要向 MIDlet 授予相应的安全授权（如果尚未授权）。选择“编辑”>“首选项”。单击“安全”选项卡。将安全域更改为最大值，然后按“确定”。

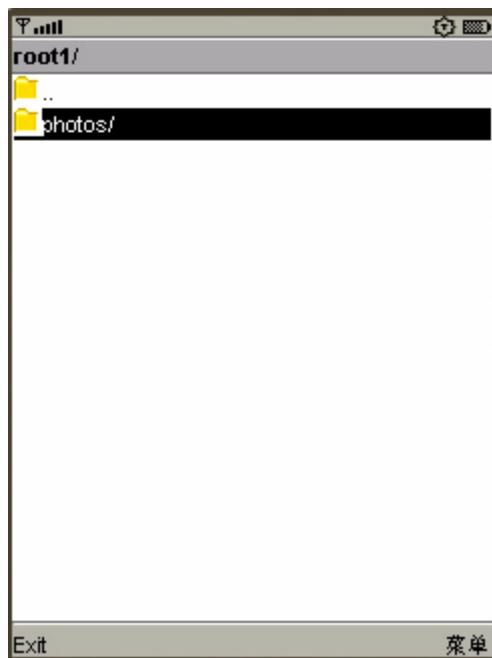
现在将打开并运行 PDAPDemo 项目。启动 FileBrowse MIDlet。用户将看到一个目录列表，可以从中浏览可用的目录和文件。默认情况下，只有一个目录 root1。

图 A-26 浏览文件



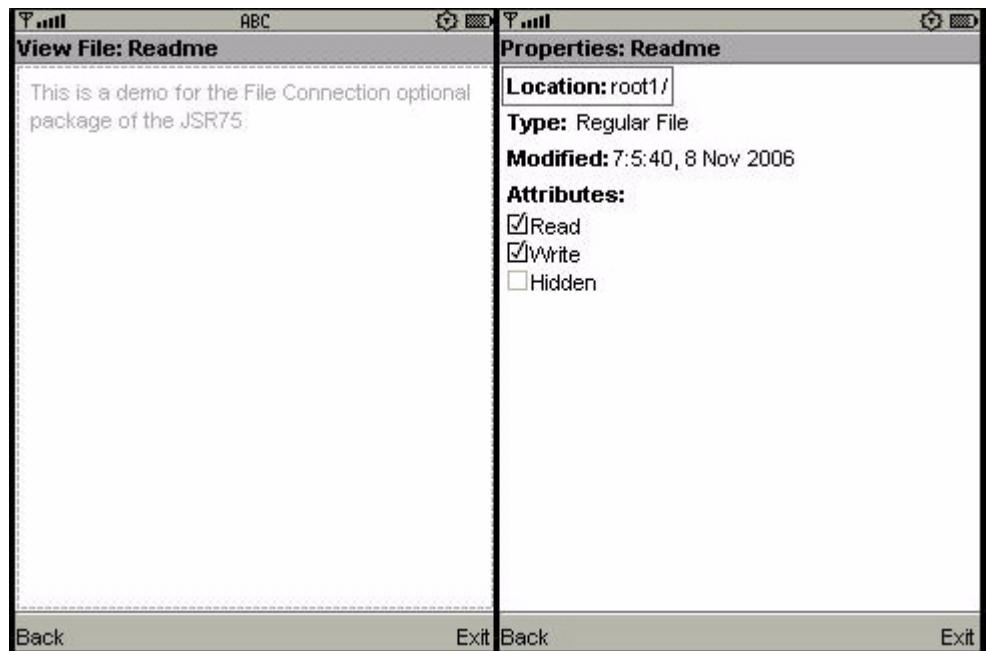
选择目录，然后按“选择”按钮输入该目录。

图 A-27 root1 目录的内容



使用演示程序中的命令，可以查看文件或其属性。请尝试选择文件，并从菜单中选择“属性”或“查看”。

图 A-28 查看文件内容和文件属性



假设使用的仿真器样机是 DefaultColorPhone，那么实际文件就位于 `workdir\appdb\DefaultColorPhone\filesystem` 中。可以根据需要添加文件和根目录，它们对于 JSR 75 File API 将是可见的。有关更多信息，请参见第 10 章。

A.16.2 PIM API

JSR75 PIM APIs 示例演示如何访问个人信息，如联系人列表、日历和任务列表。启动此示例后，从主菜单中选择列表类型。

在该示例应用程序中，各列表类型的工作方式相同，且每个列表类型只包含一个列表。例如，如果选择 "Contact Lists"，则只有一个名为“联系人”的联系人列表。"Event Lists" 中只包含一个名为“事件”的列表；"To Do Lists" 中只包含一个名为“任务”的列表。

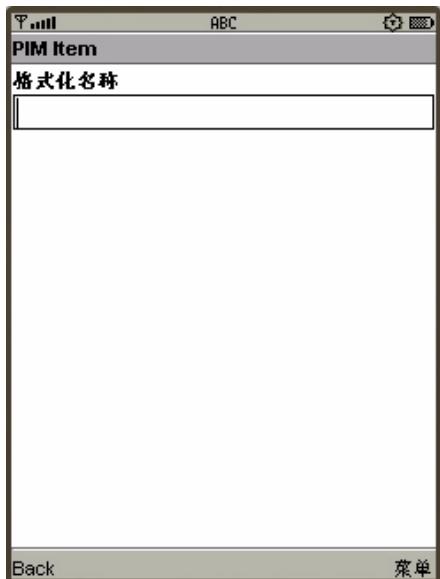
图 A-29 选择列表类型



选择列表类型和特定列表后，即可查看列表中所有的项。如果是第一次运行该示例，则列表可能是空的。

要添加项，请从菜单中选择“新建”。应用程序提示用户为该项输入一个格式化名称。用户可使用菜单中的“添加字段”为该项添加更多数据字段。将看到一列字段名称。选择一个字段，然后输入该新字段的值。

图 A-30 添加联系人字段



要保存列表项，请从菜单中选择“提交”（选项 3）。

可通过选择 "Back" 命令返回列表。用户可在列表中看到刚创建的项。

创建的项以标准的 vCard 或 vCalendar 格式存储在 *workdir\appdb\skin\pim* 目录中。有关更多信息，请参见第 10 章。

PIM API 允许以标准格式导出联系人、日历和任务项。具体的格式根据列表类型而定。在查看任一列表中的项时，菜单中将包含用于查看导出项的命令。

例如，查看联系人列表项时，菜单中包含“显示 vCard”。选择此命令后，导出的项就会显示在屏幕上。日历项以及任务项都以 vCalendar 格式导出。

A.17 SATSADemos

SATSADemos 包括 Security and Trust Services API (SATSA) 的演示程序。大多数演示程序展示了如何与智能卡进行通讯。仿真器可使用套接字协议与模拟的智能卡进行通讯。工具包括智能卡仿真器 (`cref`)。有关详细信息，请参见第 14 章。

以下部分对该演示中的每个菜单选择进行了说明。对于每个演示，请确保在启动仿真器之前执行以下操作：

- 从命令行运行 `cref` 的实例。
- 确保将安全域设为最大值。

A.17.1 APDUMIDlet

该 MIDlet 演示了如何使用应用程序协议数据单元 (APDU)（小型数据包）与智能卡进行通讯。APDUMIDlet 应找到两个模拟的智能卡。用户可使用作为 Java Card Development Kit 一部分的 `cref` 运行智能卡仿真器。

Mohair 应用程序包括可以与 `cref` 一起使用的预生成内存映像。内存图像包含可与 Mohair 交互的 Java Card 应用程序。内存图像位于 Mohair 项目的根目录中。

在 Windows 上，根据如下所示启动 `cref` 的两个实例，每个模拟卡插槽对应一个实例（假设当前目录为 toolkit 安装目录）：

```
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，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
```

请注意，端口号（在此例中为 9025 和 9026）必须符合在 SATSA 首选项中指定的端口号，如第 14 章中所述。另外，请确保使用 `demo2.eeprom` 的正确路径。

这两个智能卡仿真器运行后，用户可以运行 APDUMIDlet。

A.17.2 SATMIDlet

SATMIDlet 演示了与 APDU 通讯稍有不同的智能卡通讯。

要设置模拟智能卡，请使用 `cref`，与对 APDUMIDlet 所执行的操作非常类似。这次用户不必指定端口号，并且内存图像有所不同：

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

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

智能卡仿真器运行后，用户可运行 SATMIDlet 与智能卡应用程序进行通讯。

A.17.3 CryptoMIDlet

CryptoMIDlet 演示了 SATSA 的常规加密功能。它与智能卡之间不会以任何方式进行交互。

A.17.4 MohairMIDlet

MohairMIDlet 有两个功能。第一个功能是查找插槽，显示所有可用的卡插槽。每个插槽都有一个后跟 ‘C’ 或 ‘H’ 的数字，表示该插槽是冷插拔还是热插拔。查看插槽后，请选择“返回”返回到第一个屏幕。

MohairMIDlet 的第二部分是 SATSA-PKI 签名测试，使用智能卡生成数字签名。与前面的演示程序一样，用户需要通过正确的内存图像运行 cref，以便准备与 MohairMIDlet 进行连接。在安装目录中键入以下内容：

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

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

在仿真器中，高亮显示 SATSA-PKI 签名测试，然后选择“选择”按钮。将显示以下确认消息：

This certificate will be used: MohairAuth

选择“确定”软键。

对于 PIN 1，请键入：1234

选择“确定”软键。将显示以下确认消息：

This string will be signed: JSR 177 Approved

选择“确定”软键。将显示以下确认消息：

This certificate will be used: MohairAuth

选择“确定”软键。

对于不可否认性键 1 PIN，请键入：2345

A.18 SATSAJCRMIDemo

该应用程序包含一个 MIDlet (JCRMIMIDlet)，它展示了如何使用 Java Card RMI（卡友好远程对象协议）与卡应用程序进行通讯。与 SATSADemos 中的某些 MIDlet 一样，用户需要使用相应的内存图像启动 cref：

对于 Windows 系统： start bin\cref -p 9025 -i apps\SATSADemos\demo2.eeprom

对于 Linux 系统： workdir/bin/cref -i apps/SATSADemos/demo2.eeprom

现在，运行 JCRMIMIDlet 以查看应用程序如何与卡上的分布式对象进行通讯。

A.19 SIPDemo

该应用程序是使用 SIP (JSR 180) 在两个设备之间直接进行通讯的简单示例。通常，设备会使用具有代理服务器的 SIP 来设置某种直接的通讯。有关代理的更完整示例，请查看 GoSip。

要查看 SIPDemo 如何工作，请运行仿真器的两个实例。在第一个仿真器中，选择“接收消息”。可以使用默认端口 5070，然后选择“接收”。第一个仿真器现在将侦听传入消息。

在第二个仿真器中，选择“发送消息”。填写收件人、端口号、主题和消息的值，或接受默认值，然后选择“发送”。用户的消息将显示在第一个仿真器中。第一个仿真器的响应将显示在第二个仿真器中。

请在打开网络监视器后再次尝试。用户可在网络监视器 "SIP" 选项卡中查看仿真器之间的通讯。

A.20 SVGContactList

该应用程序使用不同的样机来显示相同的联系人列表信息和新闻标题。这些样机有不同的颜色和字体。

选择 "SVGContactlist(skin 1)" 或 "SVGContactlist(skin 2)"，然后单击“启动”。

使用上箭头和下箭头在联系人列表中进行导航。高亮显示的名称将标有特殊字符 (> 或圆点)，并以较大的字体显示。

图 A-31 使用样机 2 显示联系人列表



按“选择”按钮查看高亮显示的名称的详细信息。

图 A-32 联系人列表详细信息



再次按“选择”按钮返回到联系人列表。

A.21 SVGDemo

该套件包含 MIDlet，演示了 JSR 226 Scalable 2D Vector Graphics API for J2ME 的不同使用方式。该 API 提供了装入、操纵、渲染和播放 SVG 内容的多种方法。

Scalable Vector Graphics (SVG) 1.1 规范定义了用于描述 XML 格式的二维图形的语言。可在网址 <http://www.w3.org/TR/SVG11/> 获取完整规范。

SVG Tiny (SVGT) 属于 SVG 的一部分，适用于诸如移动电话等小型设备。请参见 <http://www.w3.org/TR/SVGMobile/>。SVG Tiny 是功能强大的压缩 XML 格式，用于描述丰富的交互式二维动画内容。图形元素可以通过 SVG 标记进行逻辑分组和标识。

A.21.1 SVG 浏览器

SVGBrowser MIDlet 显示位于电话文件系统中的 SVG 文件。运行该演示之前，请在目录 `workdir\appdb\DefaultColorPhone\filesystem\root1` 中放置 SVG 文件。

启动演示。应用程序将显示 `root1` 的内容。选择 SVG 文件，然后选择“打开”软键。

A.21.2 渲染 SVG 图像

“渲染 SVG 图像”将装入文件中的 SVG 图像并渲染该图像。查看演示代码就会看出图像大小是即时调整，以准确符合显示区域。输出十分清晰准确。

A.21.3 播放 SVG 动画

该应用程序播放一段描述万圣节贺卡的 SVG 动画。按 8 播放，按 5 开始，按 0 停止。如果按 8，请按 5 恢复动画。如果按 0，请按 5 从头开始播放动画。

SVG 文件描述各种图像元素是如何随时间变化以构成该动画短片的。

在以下代码样例中，使用 JSR 226 `javax.microedition.m2g.SVGImage` 类装入了 SVG 资源，`javax.microedition.m2g.SVGAnimator` 类就可以处理所有复杂的 SVG 动画，并提供播放动画的 `java.awt.Component` 或 `javax.swing.JComponent`。`SVGAnimator` 类提供了播放、暂停和停止动画的方法。

```
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 从头开始创建 SVG 图像

该演示使用 API 调用生成图像。其会创建空 `SVGImage`，然后使用图形内容填充该 `SVGImage`，然后显示该内容。

A.21.5 弹力球

“弹力球”将播放一段 SVG 动画。按 8 播放，按 5 开始，按 0 停止。如果按 8，请按 5 恢复动画。如果按 0，请按 5 从头开始播放动画。

A.21.6 优化的菜单

在该演示中，选择的图标将显示黄色边界。移动到新图标后，该图标将变为选定状态，并且上一个图标将变为未选定状态。如果导航出图标格线，选择将进行循环。也就是说，如果选择了行中的最后一个图标，则向右移动会选择同一行中的第一个图标。

该演示演示了将 UI 标记与 Java 相结合所带来的灵活性：丰富的功能集（图形、动画和高端二维渲染）和在图形操纵、预渲染或播放方面的灵活性。

在此例中，图形艺术家提供的 SVG 动画定义了菜单图标的转换状态（从未选定状态变为选定状态）。该程序使用 JSR 226 API 将每个图标的动画序列分别渲染到后台屏幕缓冲区中（用于以后进行快速渲染）。

通过进行缓冲，MIDlet 可以适合设备显示屏的分辨率（因为图形是以 SVG 格式定义的），并仍保持位图渲染速度。另外，MIDlet 还会使用 SVG 的动画功能。

定义菜单项外观及其动画效果的任务（图形艺术家和设计师的工作）和显示菜单并根据菜单选择开始操作的任务（开发者的工作）划分极为清楚。只要艺术家和开发者都遵守 SVG 文档结构约定，这两项任务便可独立区分。

A.21.7 图像装饰器

在该演示中，用户可使用电话键向照片中增加装饰。键值为：

-
- 1 缩小键
 - 2 下一张图像键
 - 3 扩大键
 - 4 帮助键
 - 5 水平翻转键
 - 6 垂直翻转键
 - 7 逆时针旋转键
 - 8 上一张图像键
 - 9 顺时针旋转键
 - # 显示检出器选项
-

该演示提供了 16 个图像供用户进行装饰。

使用 2 和 6 键前后翻页来查看照片。

要装饰照片，请按 # 键显示检出器。使用箭头键高亮显示图形对象。将扩大高亮显示的对象。按“选择”按钮选择当前图形或按箭头键以高亮显示其他图形。再次按“选择”按钮向照片中添加图形。添加装饰后，用户可在图形上看到红色的 +。这表示已选择该图形，并可移动、操纵该图形以及调整图形大小。

图 A-33 使用精选语录装饰图像



使用导航箭头移动图形。使用 1 缩小图形，使用 3 扩大图形。使用 5 或 6 翻转，使用 7 或 9 旋转。对图形位置满意后，请按“选择”按钮。请注意，将显示绿色的三角形。这是光标。使用导航键在图像周围移动绿色三角形。当光标移动到对象上时，将使用一个红色框高亮显示该对象。按“选择”按钮。红色的 + 表示已选定该对象。

图 A-34 高亮显示的胡须



要删除装饰（属性），请选择对象，然后单击“菜单”软键。按 2 删除属性。

A.21.8 定位服务

启动应用程序。将显示闪现屏幕（还用作帮助）。初始视图是旧金山漫步的路线地图。海湾（蓝色）位于屏幕右侧。按 1 开始以下路线。应用程序将放大用户在地图上的位置。路线指引将显示在水平轴上的白色框中。路线运行时，按 7 逆时针旋转地图。请注意，地图将旋转并且文字现在将显示在垂直轴上。再次按 7 恢复默认方位。按 4 显示帮助屏幕。

图 A-35



A.22 WMADemo

该应用程序展示如何发送和接收 SMS、CBS 和 MMS 消息。Sun Java™ Wireless Toolkit for CLDC 提供了灵活的仿真环境来支持消息传送。可以在仿真器实例之间交换消息，并可使用 WMA 控制台实用程序生成或接收消息。

因为该示例使用推送注册表，因此仅通过使用“运行”按钮无法查看所有功能。使用“通过 OTA 运行”功能可将应用程序安装到仿真器上，该安装过程与在真实设备上安装应用程序的过程相同。如果要了解具体方法，请阅读第 2 章中的介绍。

要练习如何推送注册表，请使用 WMA 控制台向仿真器发送消息。通过选择“文件”>“实用程序”启动控制台。单击 WMA 框中的“打开控制台”按钮启动 WMA 控制台。

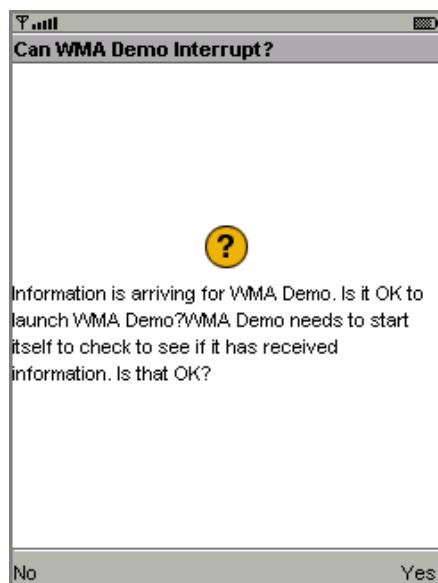
单击 WMA 控制台窗口中的“发送 SMS...”按钮。选择仿真器对应的号码（如 +5550000），如果不能确定仿真器使用的号码，可以在标题栏中找到。在 SMS 消息窗口中选择号码，然后填写端口号 50000。在“消息”字段中键入文本消息，然后单击“发送”。

图 A-36 发送文本消息



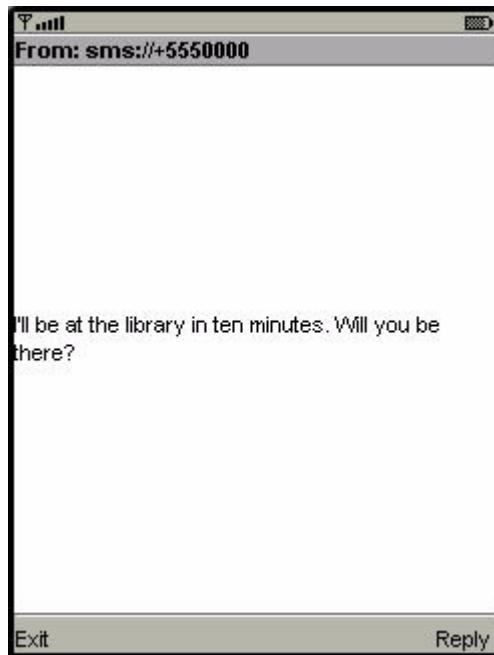
仿真器将询问是否可以启动 WMADemo 应用程序。

图 A-37 推送注册表消息



选择 "Yes"。 SMSReceive MIDlet 启动并立即显示传入的 SMS 消息。

图 A-38 传入消息文本



用户还可以使用 WMA 控制台发送和接收 CBS 和 MMS 消息。有关更多信息，请参见第 7 章。

注 - 如果尝试使用 WMA 控制台向 WMADemo 发送文本消息，请确保将端口号指定为 50000，并将端口 50001 用于 CBS 消息。对于 MMS 消息，请将 example.mms.MMSDemo 用作应用程序 ID。

例如，要从 WMA 控制台向仿真器发送 MMS 消息，请确保已经按照上述说明使用“通过 OTA 运行”安装了 WMADemo。启动演示，然后选择“MMS 接收”。

在 WMA 控制台中，单击“发送 MMS...”打开 MMS 编写窗口。填写消息主题、应用程序 ID example.mms.MMSDemo 以及正在运行的仿真器的电话号码。

图 A-39 填写 MMS 消息地址



接下来，单击“部分”选项卡。WMA 控制台允许您从硬盘选择要作为 MMS 消息的一部分进行发送的文件。单击“添加”向消息中添加文件。使用文件浏览器查找想要发送的文件，然后单击“确定”。

图 A-40 向 MMS 消息中添加部分



单击“发送”发送消息。

仿真器将询问是否可以启动 WMADemo。单击 "Yes"。显示图像及其信息。

图 A-41 WMADemo 接收图像



命令行参考

本附录介绍如何通过命令行运行 Sun Java™ Wireless Toolkit for CLDC，并详细说明生成和运行应用程序所需执行的步骤。本附录还介绍称为 MEKeyTool 的 Sun Java™ Wireless Toolkit for CLDC 证书管理器实用程序，以及称为 JadTool（Java 应用程序描述符工具）的 MIDlet 签名实用程序。

B.1 先决条件

通过命令行生成和运行应用程序之前，请验证 Java SE 软件开发工具包的版本是否不低于 1.4.2。请确保 jar 命令位于路径中。要查看开发工具包的版本，请运行 jar 命令，然后在命令行中运行 java -version。

有关更多示例，请参见位于演示应用程序 bin 目录下的文件 build.bat 和 run.bat。可以在以下位置找到这些文件：

对于 Windows 系统： toolkit\apps\demo\bin

对于 Linux 系统： toolkit/apps/demo/bin

toolkit 是 Sun Java™ Wireless Toolkit for CLDC 的安装目录， demo 是其中一个演示应用程序的名称。

B.2 开发周期

有关开发 MIDP 应用程序的完整说明, 请参见[第 2 章](#)。本节介绍如何从命令行中完成开发周期的各个步骤。

B.2.1 生成

在用户界面中, 生成项目只需一步。但实际上需要执行两个步骤。首先, 将 Java 源文件编译到 Java 类文件中。接下来将预校验这些类文件, 以确定其是否已为 CLDC KVM 做好准备。

使用 Java SE 开发工具包中的 `javac` 编译器编译 Java 源文件。可以使用现有的 Sun Java™ Wireless Toolkit for CLDC 项目目录结构。用户需使用 `-bootclasspath` 选项通知编译器要使用 MIDP API, 并使用 `-d` 选项通知编译器放置编译完的类文件的位置。

以下示例说明如何编译 MIDP 2.0 应用程序, 即从 `src` 目录下取得源文件, 然后将类文件放置在 `tmpclasses` 目录下。为了清晰起见, 进行换行显示。

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
```

如果要使用工具包支持的可选 API, 请将这些 API 的 JAR 文件添加到 `-bootclasspath` 选项中。

有关 `javac` 的更多信息, 请参阅 Java SE 文档。

下一步是预校验类文件。Sun Java™ Wireless Toolkit for CLDC 的 `bin` 目录下, 有一个名为 `preverify` 的便利实用程序。预校验命令的语法如下:

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

其中某些选项如下：

`-classpath classpath`

指定装入类的目录或 JAR 文件（以分号分隔列表的形式提供）。

`-d output-directory`

指定输出类的目标目录。该目录必须在预校验前就已存在。如果不使用该选项，预校验器会将类置于名为 `output` 的目录下。

请按照示例进行编译，使用以下命令验证已编译的类文件。与前面一样，为了清晰起见，进行换行显示。

Windows

```
preverify
```

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

Linux

```
preverify
```

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

执行该命令后，会将预校验的类文件置于 `classes` 目录下。如果应用程序使用的是 WMA、MMAPI 或其他版本的 CLDC 或 MIDP，请确保在类路径中包含相关的 `.jar` 文件。

B.2.2 打包

要将 MIDlet 套件打包，必须创建清单文件、应用程序 JAR 文件，最后还要有一个 MIDlet 套件描述符。

创建包含 MIDP 规范中所指定的相应属性的清单文件。可以使用任何文本编辑器创建清单文件。清单可能包含下列内容，例如：

```
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
```

创建包含清单文件以及套件的类和资源文件的 JAR 文件。要创建 JAR 文件，请使用 Java SE 软件开发工具附带的 `jar` 工具。语法如下：

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

参数如下：

- *file*——要创建的 JAR 文件。
- *manifest*——MIDlet 的清单文件。
- *class-directory*——包含应用程序类的目录。
- *resource-directory*——包含应用程序资源的目录。

例如，要创建名为 `MyApp.jar` 的 JAR 文件（此文件的类位于 `classes` 目录下，资源位于 `res` 目录下），请使用以下命令：

```
jar cfm MyApp.jar MANIFEST.MF -C classes. -C res .
```

创建包含 MIDP 规范中所指定的相应属性的 JAD 文件。可以使用任何文本编辑器创建 JAD 文件。该文件的扩展名必须为 `.jad`。

注 - 用户需要将 `MIDlet-Jar-Size` 条目设置为在上一步骤中创建的 JAR 文件的大小。

例如，JAD 文件可能具有以下内容：

```
MIDlet-Name: MyMIDlet
MIDlet-Vendor: My Organization
MIDlet-Version: 1.0
MIDlet-Jar-URL: MyApp.jar
MIDlet-Jar-Size: 24601
```

B.2.3 运行

可以通过命令行运行仿真器。Sun Java™ Wireless Toolkit for CLDC 的 `bin` 目录包含 `emulator` 命令。`emulator` 命令的语法如下：

```
emulator options
```

常规选项如下：

- `-help`——显示有效选项列表。
- `-version`——显示仿真器的版本信息。
- `-Xquery`——按标准输出流方式打印仿真器样机信息并立即退出。这些信息包括样机名称、屏幕大小以及其他功能。

专门用于运行 MIDlet 套件的选项如下：

- **-Xdevice:skin-name**——使用给定的样机名称在仿真器上运行应用程序。有关样机名称列表，请参见第 4.1 节，第 4-1 页上的“仿真器样机”。
- **-Xdescriptor:jad-file**——使用给定的 JAD 文件在本地运行应用程序。
- **-classpath classpath**——指定运行应用程序所需的库的类路径。请在从本地运行应用程序时使用此选项。
- **-Dcom.sun.midp.io.http.proxy**——运行时设置 HTTP 和 HTTPS 代理服务器。例如：

-Dcom.sun.midp.io.http.proxy=*proxy-host:proxy-port*

- **-Dcom.sun.midp.midlet.platformRequestCommand**——指定应用程序呼叫 URL 时要使用的浏览器。例如：

-Dcom.sun.midp.midlet.platformRequestCommand=*browser*

样例呼叫可能类似于：

```
emulator -Dcom.sun.midp.midlet.platformRequestCommand=firefox  
-Xjam:install=URL-to-app-using-platformRequest-method
```

如果希望每次均使用相同的浏览器，则可以将以下行添加到 *toolkit/lib/system.config*:

com.sun.midp.midlet.platformRequestCommand: 浏览器

Windows: 在 Windows 中，如果未指定此参数，则将使用默认浏览器。

Linux: 对于 Linux，则必须指定此参数，因为 Linux 系统通常不具有默认浏览器。如果未指定此参数，则应用程序尝试打开 URL 时，将无任何反应。

- **-Xjam:command=application**——使用应用程序管理软件 (Application Management Software, AMS) 远程运行应用程序，以使用 OTA 置备进行运行。如果未使用参数指定应用程序，则会运行图形式 AMS。命令如下：

install=jad-file-url | force | list | storageNames |

将带有指定 JAD 文件的应用程序安装到设备上。

此外

run= [storage-name | storage-number]

运行先前安装的应用程序。可通过应用程序的有效存储名称或存储号码来指定应用程序。

remove= [storage-name | storage-number | all]

删除先前安装的应用程序。可通过应用程序的有效存储名称或存储号码来指定应用程序。如果指定 all，将删除所有先前安装的应用程序。

- **transient=jad-file-url**——安装、运行和删除带有指定 JAD 文件的应用程序。如果指定 transient，则会执行安装应用程序，运行该应用程序，然后将其删除这三步。

B.2.4 调试

可以对仿真器使用下列选项来进行调试和追踪。

- `-Xverbose : trace-options`——显示追踪输出，如逗号分隔的选项列表所指定的，如下：
 - `gc`——追踪垃圾回收
 - `class`——追踪类装入
 - `all`——使用所有追踪选项
- `-Xdebug`——启用运行时调试。还必须使用 `-Xrunjdwp` 选项。
- `-Xrunjdwp : debug-settings`——启动 Java 调试有线协议会话，如逗号分隔的调试设置列表所指定。还必须使用 `-Xdebug` 选项。有效的调试设置包括以下内容：
 - `transport=transport-mechanism`——用来与调试器进行通讯的传输机制。唯一支持的传输机制是 `dt_socket`。
 - `address=host:port`——调试器连接的传输地址。可以不必提供 `host`。如果省略 `host`，则假定 `localhost` 为主机。
 - `server={y|n}`——启动作为服务器的调试代理。调试器必须连接到指定的端口。可能的值为 `y` 和 `n`。当前仅支持 `y`（仿真器必须充当服务器）。

B.3 启动工具包 GUI 组件

Sun Java™ Wireless Toolkit for CLDC 的组件均可从命令行启动。每个组件均位于工具包的 `bin` 目录下。

表 B-1 工具包组件命令

| 命令 | 描述 |
|----------------------------|--------------------|
| <code>DefaultDevice</code> | 显示对话框，供用户选择默认仿真器样机 |
| <code>ktoolbar</code> | 启动用户界面。 |
| <code>prefs</code> | 启动工具包首选项 |
| <code>utils</code> | 启动工具包实用程序窗口 |

B.4 设置仿真器首选项

用户可以通过使用 `-emulator` 命令的 `--Xprefs` 选项从命令行更改仿真器首选项。格式如下：

`-Xprefs:filename`

提供作为属性文件的完整路径名的 *filename*，其值将覆盖首选项对话框中的值。属性文件可以包含下表中描述的属性。

表 B-2 仿真器首选项属性列表

| 属性名 | 属性描述和合法值 |
|--|---|
| <code>http.version</code> | 网络配置 > HTTP 版本 值：HTTP/1.1 HTTP/1.0 |
| <code>http.proxyHost</code> | 网络配置 > HTTP 地址 值：主机名 |
| <code>http.proxyPort</code> | 网络配置 > HTTP 端口 值：整型 |
| <code>https.proxyHost</code> | 网络配置 > HTTPS 地址 值：主机名 |
| <code>https.proxyPort</code> | 网络配置 > HTTPS 端口 值：整型 |
| <code>kvem.memory.monitor.enable</code> | 监视器 > 启用内存监视器 值：true false |
| <code>kvem.netmon.comm.enable</code> | 监视器 > 启用 Comm 监视 值：true false |
| <code>kvem.netmon.datagram.enable</code> | 监视器 > 启用数据报监视 值：true false |
| <code>kvem.netmon.http.enable</code> | 监视器 > 启用 HTTP 监视 值：true false |
| <code>kvem.netmon.https.enable</code> | 监视器 > 启用 HTTPS 监视 值：true false |
| <code>kvem.netmon.socket.enable</code> | 监视器 > 启用套接字监视 值：true false |
| <code>kvem.netmon.ssl.enable</code> | 监视器 > 启用 SSL 监视 值：true false |

表 B-2 仿真器首选项属性列表（续）

| 属性名 | 属性描述和合法值 |
|-------------------------------|--|
| kvem.profiler.enable | 监视器 > 启用 Profiling 值: true false |
| netspeed.bitpersecond | 性能 > 位 / 秒组合框 值: 整型 |
| netspeed.enableSpeedEmulation | 性能 > 启用网络吞吐量仿真 值: true false |
| screen.graphicsLatency | 性能 > 图形图元时延 值: 整型 |
| screen.refresh.mode | 性能 > 显示刷新 (单选按钮) 值: 默认 立即 定时 |
| screen.refresh.rate | 性能 > 显示刷新 (滑块) 值: 整型 |
| vmspeed.bytecodespermilli | 性能 > 启用虚拟机速度仿真 (复选框) 值: 整型 |
| vmspeed.enableEmulation | 性能 > 启用虚拟机速度仿真 (滑块) 值: true false |
| storage.root | 存储 > 存储根目录 值: 字符串 (<i>appdb</i> 的相对路径) |
| storage.size | 存储 > 存储大小 值: 整型 |
| mm.control.capture | MMedia > 音频捕获 值: true false |
| mm.control.midi | MMedia > MIDI 音调 值: true false |
| mm.control.mixing | MMedia > 音频混合 值: true false |
| mm.control.record | MMedia > 音频录音 值: true false |
| mm.control.volume | 值: true false |
| mm.format.midi | MMedia > MIDI 格式 值: true false |
| mm.format.video | MMedia > 视频格式 值: true false |

表 B-2 仿真器首选项属性列表（续）

| 属性名 | 属性描述和合法值 |
|-------------------------------------|--------------------------------------|
| mm.format.wav | MMedia > WAV 音频格式 值: true false |
| wma.client.phoneNumber | WMA > 下一个仿真器的电话号码 值: 整型 |
| wma.server.firstAssignedPhoneNumber | WMA > 首次指定的电话号码 值: 整型 |
| wma.server.percentFragmentLoss | WMA > % 随机消息片段丢失 值: 整型 |
| wma.server.deliveryDelayMS | WMA > 消息片段发送延迟（毫秒） 值: 整型 |

B.5 使用安全功能

通过命令行还可以使用 Sun Java™ Wireless Toolkit for CLDC 的所有安全功能。用户可以调整仿真器的默认保护域、签名 MIDlet 套件以及管理证书。

B.5.1 更改仿真器的默认保护域

要调整仿真器的默认保护域，请使用 `emulator` 命令的下列选项：

`-Xdomain domain-type`

为 MIDlet 套件指定安全域。域类型包括 `untrusted`、`trusted`、`minimum` 和 `maximum`。

B.5.2 签名 MIDlet 套件

JadTool 是用于根据 MIDP 2.0 规范使用公钥密码学签名 MIDlet 套件的命令行接口。签名 MIDlet 套件是指向 JAD 文件添加签名者证书和 JAR 文件的数字签名的过程。JadTool 还可以签名付费更新 (JPP) 文件。

JadTool 仅使用 Java SE 平台密钥库中的证书和密钥。Java SE 软件提供用于管理 Java SE 平台密钥库的命令行工具 `keytool`。

JadTool 打包在 JAR 文件中。要运行该工具，请打开命令提示符，将当前目录更改为 toolkit\bin，然后输入以下命令：

```
java -jar JadTool.jar command
```

命令如下：

- **-help**

打印 JADTool 的使用指导。

- **-addcert -alias alias [-keystore keystore] [-storepass password] [-certnum number] [-chainnum number] [-encoding encoding] -inputjad | inputjpp input-file -outputjad | outputjpp output-file**

将给定密钥库中的密钥对证书添加到 JAD 文件或 JPP 文件中。

- **-addjarsig -jarfile jarfile -keystore keystore -alias alias -storepass password -keypass password -inputjad input-jadfile -outputjad output-jadfile**

将给定 JAR 文件的数字签名添加到指定的 JAD 文件中。-jarfile 的默认值为 JAD 文件中的 MIDlet-Jar-URL 属性。

- **-showcert [([-certnum number] [-chainnum number]) | -all [-encoding encoding] -inputjad filename**

显示给定 JAD 文件中的证书列表。

- **-addjppsig -alias alias -keypass password [-keystore keystore] [-storepass password] [-encoding encoding] -inputjpp filename -outputjpp filename**

将输入 JPP 文件的数字签名添加到指定的输出 JPP 文件中。

默认值如下：

- **-encoding——UTF-8**

- **-jarfile——JAD 文件中的 MIDlet-Jar-URL 属性**

- **-keystore——%HOMEPATH%\keystore**

- **-certnum——1**

- **-chainnum——1**

B.5.3 管理证书

MEKeyTool 用于管理证书授权机构 (CA) 的公钥，这在功能上类似于 Java SE SDK 附带的实用程序 keytool。可以使用密钥来增强跨越 SSL 的安全 HTTP 通讯 (HTTPS)。

使用 MEKeyTool 之前，必须首先具有对 Java 加密扩展密钥库的访问权限。可使用 Java SE keytool 实用程序创建一个。

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>

要运行 MEKeyTool，请打开命令提示符，将当前目录更改为 *toolkit\bin*，然后输入以下命令：

对于 Windows 系统: *toolkit\bin\mekeytool.exe -command*

对于 Linux 系统: *toolkit/bin/mekeytool -command*

接下来会出现命令关键字。请注意，当 MEKeyTool 从安装目录中运行时，默认密钥及您所创建的密钥将位于您的个人目录 *workdir\appdb* 中。

■ **-help**

打印 MEKeyTool 的使用指导。

■ **-import -alias alias [-keystore JCEkeystore] [-storepass storepass] -domain domain-name**

使用给定 Java 加密扩展密钥库口令将给定的 JCE 密钥库中的公钥导入到 ME 密钥库中。默认的 ME 密钥库为 *workdir\appdb_main.mks*，默认的 Java 加密扩展密钥库为 *user.home\.keystore*。

■ **-list**

列出 ME 密钥库中的密钥，包括每个密钥的所有者及有效期。ME 密钥库为 *workdir\appdb_main.mks*。

■ **-delete (-owner owner | -number key-number)**

从给定的 ME 密钥库中删除指定所有者的密钥。ME 密钥库为 *workdir\appdb_main.mks*。

注 – Sun Java™ Wireless Toolkit for CLDC 包含名为 *_main.mks* 的 ME 密钥库，该密钥库位于 *appdb* 子目录下。该密钥库中包含 Java SE SDK 安装中附带的默认 Java SE 平台密钥库中的所有证书。

B.6 使用桩模块生成器

移动客户机可使用桩模块生成器访问 Web 服务。`wscompile` 工具可生成用于 Java API for XML (JAX) RPC 客户机和服务的桩模块、绑定、串行器和 WSDL 文件。该工具读取指定 WSDL 文件、模型文件或编译的服务端点接口的配置文件。桩模块生成器命令的语法如下：

```
wscompile [options] configuration-files
```

B.6.1 选项

表 B-3 `wscompile` 命令的选项

| 选项 | 描述 |
|-----------------------------------|---------------------------------------|
| <code>-d output directory</code> | 指定存放生成的输出文件的位置 |
| <code>-f :features</code> | 启用给定的功能 |
| <code>-features:features</code> | 与 <code>-f :features</code> 相同 |
| <code>-g</code> | 生成调试信息 |
| <code>-gen</code> | 与 <code>-gen:client</code> 相同 |
| <code>-gen:client</code> | 生成客户机工件（如桩模块等） |
| <code>-httpproxy:host:port</code> | 指定 HTTP 代理服务器（默认端口为 8080） |
| <code>-import</code> | 仅生成接口和值类型 |
| <code>-model file</code> | 将内部模型写入给定的文件 |
| <code>-O</code> | 优化已生成的代码 |
| <code>-s directory</code> | 指定存放生成的源文件的位置 |
| <code>-verbose</code> | 输出消息，说明编译器正在执行的操作 |
| <code>-version</code> | 打印版本信息 |
| <code>-cldc1.0</code> | 将 CLDC 版本设置为 1.0（默认值）。浮点型和双精度型变为字符串型。 |
| <code>-cldc1.1</code> | 将 CLDC 版本设置为 1.1（可以是浮点型和双精度型） |
| <code>-cldc1.0info</code> | 显示所有 CLDC 1.0 信息和警告消息。 |

注 - 必须仅指定一个 `-gen` 选项。 `-f` 选项要求提供功能的逗号分隔列表。

表 B-4 列出 -f 选项所包含的功能（逗号分隔）。wscompile 工具可读取作为输入的 WSDL 文件、已编译的服务端点接口 (Service Endpoint Interface, SEI) 或模型文件。“文件类型”列指出可与某项特定功能一起使用的文件。

表 B-4 wscompile 的命令支持的功能 (-f)

| 选项 | 描述 | 文件类型 |
|---------------------|---------------------|-------------|
| explicitcontext | 打开显式服务上下文映射 | WSDL |
| nodatabinding | 关闭文字编码的数据绑定 | WSDL |
| noencodedtypes | 关闭编码类型信息 | WSDL、SEI、模型 |
| nomultirefs | 关闭对多重引用的支持 | WSDL、SEI、模型 |
| novalidation | 关闭已导入 WSDL 文档的完整验证 | WSDL |
| searchschema | 对模式进行侵入式搜索以查找子类型 | WSDL |
| serializeinterfaces | 打开接口类型的直接串行化 | WSDL、SEI、模型 |
| wsi | 启用 WSI 基本配置文件功能（默认） | |
| resolveidref | 解析 xsd:IDREF | |
| nounwrap | 不展开。 | |

示例

```
wscompile -gen -d generated config.xml
wscompile -gen -f:nounwrap -O -cldc1.1 -d generated config.xml
```


本地化

本附录介绍 Sun Java™ Wireless Toolkit for CLDC 中显示语言的设置和仿真环境中的本地化设置。

C.1 语言环境设置

语言环境是指拥有相同语言、习俗或文化传统的地理位置、行政区域或社区。在软件中，语言环境由文件、数据和代码的集合表示，它包括使软件适应特定区域所必需的信息。

某些软件使用语言环境为用户定制信息，如：

- 显示给用户的消息
- 使用的字体或其他特定于书面的信息

默认情况下，用户界面中的所有字符串均以所支持平台语言环境的语言显示。

例如，如果已经下载并安装了正确本地化的 Sun Java™ Wireless Toolkit for CLDC，则在日语版 Microsoft Windows 计算机上运行的工具包中可以显示日语字符。

可以设置 `wtk.locale` 属性，使用户界面以指定的语言环境的语言显示。如[第 3.8 节](#)，[第 3-13 页上的“配置 Wireless Toolkit”](#)中所述，您必须将 `ktools.properties` 从安装目录复制到您的工作目录 (`workdir/wtklib/ktools.properties`)，然后编辑副本。例如，通过将语言环境属性设置为 `en-US`，并确保下载了适当的增补程序并将其安装在 Sun Java™ Wireless Toolkit for CLDC 上，可以使该工具包运行在日语版计算机上的同时仍使用户界面以英语显示。

C.2 仿真的语言环境

设备的语言环境包含在系统属性 `microedition.locale` 中。您可以通过选择“编辑”>“首选项”，然后选择“国际化”更改仿真器的语言环境。从组合框中选择语言环境或直接键入语言环境。

有关 `microedition.locale` 的信息，请查询 MIDP 规范。

C.3 字符编码

CLDC 系统属性 `microedition.encoding` 定义 MIDP 环境的默认字符编码名称。在 Sun Java™ Wireless Toolkit for CLDC 仿真器中，该属性是根据正在使用的基础视窗系统进行设置的。该属性值将被设置为在同一视窗系统上运行的 Java SE 平台的默认编码。例如，在英语视窗系统中，编码设置为：

```
microedition.encoding=ISO8859_1
```

可通过将 `microedition.encoding` 属性添加到 `workdir\wtklib\ktools.properties` 文件来覆盖默认值。例如，如果希望将 UTF-8 作为 Microsoft Windows 上的默认设置，则可按如下方式设置 `workdir\wtklib\ktools.properties` 文件中的属性：

```
microedition.encoding=UTF-8
```

有关字符编码的更多信息，请参见 CLDC 规范。

注 - 所有 Java SE 平台编码器在仿真环境中均可用。有关如何限制特定设备的可用编码器列表的信息，请参见 Sun Java™ Wireless Toolkit for CLDC 基本自定义指南。

C.4 Java 技术编译器编码设置

`javac.encoding` 属性决定了 `javac` 编译器编译源文件所使用的编码。该属性值将被设置为在同一视窗系统上运行的 Java SE 平台的默认编码。

可通过将 `javac.encoding` 属性添加到 `ktools.properties` 文件来覆盖默认值。例如，如果在英语系统中运行，但发现需要编译日语资源包，则可指定日语字符集，如：

```
javac.encoding=EUCJIS
```

C.5

默认仿真器中的字体支持

仿真环境中使用的默认字体是根据基础视窗系统语言环境设置的。默认情况下，MIDP 环境字体将映射到默认 Java SE 平台 Java 技术字体。这些字体通常支持当前视窗语言环境所要求的所有字符。

可以覆盖这些字体以支持不受默认字体支持的其他字符。有关如何对其进行配置的信息，请参见 Sun Java Wireless Toolkit for CLDC 基本自定义指南。

索引

符号

workdir, 1-1
代理服务器设置, A-4
-classpath 选项, B-3
-help 选项, B-4
-import 命令, B-11
-version 选项, B-4
-Xdebug 选项, B-6
-Xquery 选项, B-6
-Xrunjdwp 选项, B-6
-Xverbose 选项, B-6

英文

AMR, 8-2
AMS, 2-8
CBS 消息, 发送, 7-6
cref, 14-1
emulator 命令, B-4
FileConnection API, 10-1
FileConnection API 中的根, 10-2
in.use 文件, 4-5
IrDA, 11-1
J2ME Web 服务规范, 12-1
JAD 文件, 2-8
 MIME 类型, 2-16
 创建, 2-8
 属性, 3-3
JadTool, B-9

JAR 文件

 MIME 类型, 2-16
 创建, 2-8
 在打包, 2-8
Java 加密扩展 (Java Cryptography Extension, JCE)
 密钥库, B-11
JSR 75, 10-1
JSR 75, A-38, A-41
JSR 82, 11-1, A-6
JSR 118, 6-1
JSR 120, 7-1
JSR 135, 8-1
JSR 172, 12-1, A-30
JSR 177, 14-1
JSR 179, 13-1, A-11
JSR 180, 15-1, A-22, A-46
JSR 184, 9-1, A-19
JSR 185, 6-3
JSR 205, 7-1
JSR 211, A-7
JSR 226, 9-3, A-48
JSR 229, 16-1, A-27
JSR 234, A-4
JSR 238, 17-1, A-24
JSR 239, 9-3
JSR 248, 6-3
JTWI 保护域, 6-4

keytool 实用程序, B-11
ktools.properties, C-1
ktools.properties, 3-14
Location API, 13-1
M3G, 9-1
MEKeyTool, B-11
MIA, 17-1
microedition.encoding 属性, C-2
MIDlet
 JAR 文件, 2-8
 描述符, 2-8
 添加新, 3-5
 修改, 3-5
MIDlet 套件, 使用真实密钥进行签名, 6-6
MIDlet 套件, 签名, 6-4
MIME 类型, 2-16
MMAPI, 8-1, A-30
Mobile 3D Graphics API, 9-1
Mobile Internationalization API, 17-1
Mobile Media API, 8-1
Mobile Media API (MMAPI), 8-1
 捕获, 8-4
 格式和协议, 8-1
MSA 保护域, 6-3
OBEX, 11-1
 首选项, 11-3
 演示, A-37
OpenGL® ES, 9-3
Payment API, 16-1, A-27
PDA 可选包, 10-1
PDAP, 10-1
PIM API, 10-3
RevisionControl 属性, 3-14
run 选项, B-5
SATSA, 14-1
SATSA 演示, A-43
SIP API, 15-1
SMS 二进制消息, 发送, 7-5
SMS 文本消息, 发送, 7-4
SVG, 9-3
SVG 渲染, 9-3
USB 令牌, 6-9
Web 服务, 桩模块生成器, 12-1
Web 服务的桩模块生成器, 12-1
Web 服务规范, 12-1
Wireless Messaging API, 7-1
Wireless Toolkit
 通过命令行运行, B-1
 证书管理器实用程序, B-1
WMA, 7-1
WMA 控制台, 7-3, A-55
wscompile 工具, B-12
WSDL 文件, 12-2

B

版本控制, 3-14
保护域, 6-1
 JTWI, 6-4
 MSA, 6-3
编码, javac, 3-14
编译器编码, C-2

C

存储首选项, 4-4, 4-5

D

打包
 命令行示例, B-4
代理服务器, 4-3
电话号码, 在仿真器中设置, 7-1
堆栈, 4-5
堆栈大小, 4-4

F

方法调用图, 5-3
方法配置处理, 5-1
仿真器, 4-1
 单独运行, 4-7
 键盘快捷键, 4-3
默认保护域, B-9
默认字体支持, C-3
首选项, 4-3, B-7

性能, 4-5
样机, 4-1
语言环境, C-2
语言支持, C-1

G

个人信息管理 (Personal Information Management, PIM) API, 10-1

更新, 1-6

工具包

启动, 1-4, 2-1
应用程序目录, 3-14

工具包, 1-1

H

混淆, 2-14
安装 ProGuard, 2-14

J

检查更新, 1-6

K

开发周期
简单, 2-3
完整, 2-7
可选 API, 1-7
库, 3-11

L

蓝牙, 11-1
铃声, 8-4

浏览器

在仿真器运行时指定, B-5
在配置文件中指定, B-5

M

密钥对
创建, 6-5
导入, 6-7
密钥管理, 6-5
密钥库, JCE, B-11
描述符, 2-8
属性, 3-3

命令行操作, B-1
命令路径, B-1
目标平台, 3-2

N

内存监视器, 5-4
对象详细信息, 5-5
图, 5-4

Q

签名 MIDlet 套件, 6-4, B-9
清除 RMS, 3-11
清单文件, 创建, B-3
权限, 6-1

S

生成 (从用户界面), 2-5
生成 (命令行), B-2
事件探查器, 5-1
方法调用图, 5-3
属性, 3-3

T

调试, 2-15
从命令行, B-6
选项, B-6
通过 OTA 运行, 2-8, 6-3
通过命令行管理证书, B-11
通过命令行生成桩模块, B-12
推送注册表, 3-6

W

网络监视器, 5-7
对消息进行排序, 5-8
过滤, 5-8
与 WMA 配合使用, 7-8

X

项目, 2-1, 3-1
MIDlet, 3-5
创建, 2-2
打包, 2-8

根据 JAD/JAR 创建, 2-13
库, 3-11
生成, 2-5
属性, 3-3
推送注册表, 3-6
选择 API, 3-1
源代码, 2-2
运行, 2-6
在真实设备上部署, 2-12
消息 URL http
 //www.ietf.org/rfc/rfc3267.txt, 8-2
消息传递, 网络模拟, 7-3
消息树排序, 5-8
性能, 4-5
修订控制, 3-14
修订控制文件, 3-14
修订控制系统 (Revision Control System, RCS),
 3-14

Y
演示, A-1
 源代码, 1-2
已签名的 MIDlet 套件, 6-1
应用程序
 远程运行, 2-16
 在 Web 服务器上部署, 2-16
应用程序描述符, 2-8
应用程序目录, 设置, 3-14
永久存储, 4-4, 4-5
 清除数据库, 4-5
语言环境, C-1
预校验, 2-5, B-2
 命令行示例, B-3
源代码
 创建, 2-4
 位置, 2-2
远程部署的应用程序, 2-16

Z
在 Web 服务器上部署, 2-16
暂停和恢复, 4-7
证书导入, 6-8
证书管理, 6-7
证书管理器实用程序, B-1
支持的 API, 1-7
追踪选项, B-6
字符编码, C-2
字体支持, C-3