

UniJa 开发指南

2004-05-27 Ver2.1.2

目 录

1	用户界面规范	1 -
1.1	业务可用性概述	1 -
1.2	JAVA 应用程序用户接口	2 -
1.2.1	法律考虑 (Legal considerations)	2 -
1.2.2	应用程序的名称 (Naming)	2 -
1.2.3	应用程序的数据管理	2 -
1.2.4	版本号变化规则	2 -
1.2.5	JAD 属性	2 -
1.2.6	应用程序的打包	3 -
1.2.7	应用程序的闪屏 (Splash screen)	3 -
1.2.8	应用程序的主屏幕 (Main screen)	3 -
1.2.9	运行过程的等待处理 (Process waiting)	4 -
1.3	JAVA 用户接口特性	4 -
1.3.1	版式 (Typography)	4 -
1.3.2	图标 (Icons)	5 -
1.3.3	提示文本 (Alerting text)	6 -
1.3.4	菜单 (Menus)	6 -
1.3.5	文本输入和导航 (Text input and navigation)	7 -
1.3.6	表单	8 -
1.3.7	关键词 (Key words)	8 -
1.4	UNiJA 终端基本特性	8 -
1.4.1	数据传输协议	8 -
1.4.2	默认主页地址	8 -
1.4.3	应用索引	8 -
1.4.4	目录管理	9 -
1.4.5	系统设置和应用设置	9 -
1.4.6	与 R-UIM 卡的关系	11 -
1.4.7	用户注册	11 -
1.4.8	Flash ROM	11 -
1.4.9	RAM	11 -
1.4.10	LCD	11 -
1.4.11	声音	12 -
1.4.12	图像	12 -
1.4.13	摄像头	12 -
1.4.14	软件要求	12 -
1.4.15	用户界面与操作	13 -
1.4.16	测试证书	13 -
1.4.17	暂停	13 -
1.4.18	软键	13 -
2	开发设计指南	13 -
2.1	分辨率	13 -

2.2	色彩数量	13
2.3	明亮度	14
2.4	显示速度	14
2.5	电力供应	14
2.6	执行速度	14
2.6.1	程序执行速度	14
2.6.2	量度	14
2.6.3	图形操作	15
2.6.4	垃圾收集	15
2.6.5	多线程	16
2.7	网络速度	17
2.7.1	带宽和延迟时间	17
2.7.2	限制HTTP的往返传输	17
2.7.3	避免低效协议	17
2.8	文件大小	18
2.8.1	小尺寸设计	18
2.8.2	库	18
2.8.3	保持有限资源	18
2.8.4	考虑合并图象文件	19
2.9	资源利用	19
2.9.1	堆内存	19
2.9.2	网络	19
2.10	响应性能	20
2.10.1	进程提示	20
2.10.2	响应	20
2.11	DRM 介绍	20
2.11.1	概述	20
2.11.2	权限定义	21
2.11.3	DRM 方法	22
2.11.4	联通 DRM 系统	22
2.12	DIGITAL SIGNATURE 介绍	23
2.13	声音类型	25
2.14	操作键	26
2.14.1	数字和符号键	26
2.14.2	光标方向键	26
2.14.3	上下翻页键	26
2.14.4	清除键	26
2.14.5	终止键	26
2.14.6	管理键（快捷键）	26
2.14.7	索引键	26
2.14.8	功能软键	27
2.14.9	按键事件和 Game Actions	27

修改历史

日期	版本	修改人	注释
2004-03-11	V1.8.0	修增喜	增加了 JAD 的可选和必选属性
2004-04-22	V1.9.0	修增喜	增加了版本号变化规则
2004-04-29	V2.1.0	修增喜	增加了对菜单项、暂停及软键的要求
2004-05-11	V2.1.1	王晓海	增加了对应用的要求说明、对字体的要求；修改了对闪屏部分的要求、LCD 相关部分
2004-05-27	V2.1.2	王晓海	增加了对使用 Game Actions 的说明；修改了对获得系统设置、清除键、终止键的说明

1 用户界面规范

中国联通为了统一 JAVA 业务风格，保证 JAVA 业务的易用性，特制定本规范部分作为开发 JAVA 业务的约束条件。

本规范阐述了 JAVA 业务可用性的概念与原则，并对 JAVA 终端用户接口、JAVA 应用程序用户接口等方面作出了规定。对通过中国联通 JAVA 门户网站提供服务，以及要求中国联通提供代计费的 JAVA 应用程序，均要求遵循本规范。

1.1 业务可用性概述

可用性是业务提供的价值与用户使用业务时所需付出代价的相对大小，所谓业务提供的价值，主要体现在业务能提供给用户所需要的信息，业务能为用户解决实际问题以及业务的易用性等方面，使用业务的代价包括学习使用业务时付出的代价、用户使用业务过程中付出的代价（比如连接费用、信息费用、点击次数，输入等）。

可用性强的业务，其最根本的特点是业务给用户带来的价值大于用户在使用业务时所付出的代价，因此，在设计 JAVA 业务时，应该注意到：

- ☐ 可用性；
- ☐ 手机上网主要用于获取信息而不是浏览；
- ☐ 手机与 PC 在使用上有显著差别，不能将开发 WEB 应用的理念照搬到 JAVA 业务开发中来；
- ☐ 用户是要为使用 JAVA 业务付出时间上或金钱上的代价，因此，所设计的 JAVA 业务要尽量降低用户在使用过程中付出的代价。
- ☐ 大多数用户并不喜欢复杂的 JAVA 业务。

由此可见，当开发高可用性业务的时候必须首先明确：

- ☐ 面向什么样的用户群；
- ☐ 业务给用户解决什么问题；
- ☐ 怎样以最高的效率解决这些问题。

以下是开发 JAVA 业务时的可用性原则：

- ☐ 使用的灵活性和效率；
- ☐ 系统状态的显见性；
- ☐ 系统和真实世界间的匹配；
- ☐ 用户控制和自由；

- ☐ 风格的一致性和标准;
- ☐ 业务简单易用;
- ☐ 审美的考虑;
- ☐ 帮助用户识别、诊断和修复错误
- ☐ 帮助和文件编制
- ☐ 视觉合成

1.2 JAVA 应用程序用户接口

注意：如果需要编写 WMA 应用程序，必须先向中国联通申请 Port Number 号码段。

1.2.1 法律考虑（Legal considerations）

若您的应用程序中使用了其他公司以商标权或版权注册的内容，请务必根据您与该公司有关的许可协议妥善使用。

1.2.2 应用程序的名称（Naming）

必选：1、只允许采用中文名称（英文专有名词除外）。

- 2、论是否含图标(ICON)，名称应控制在一行显示。名称的字数与每一个目标设备的屏幕相匹配

1.2.3 应用程序的数据管理

如应用程序为过关类或累积积分类游戏，则必须具备分数记录功能，要求最少可以保存 5 条最高分记录并且允许用户输入对应每条记录的姓名。

1.2.4 版本号变化规则

版本号格式：Va.b.c

- a 指主版本号。主要标识应用程序的主体及架构的变化。
- b 指二级版本号。主要标识应用增加新的功能。
- c 指三级版本号。主要标识对应用问题（bug）的修改。

1.2.5 JAD 属性

在任何条件下必选的JAD属性：

MIDlet-Name

MIDlet-Version
MIDlet-Vendor
MIDlet-Jar-URL
MIDlet-Jar-Size
MIDlet-Description

1.2.6 应用程序的打包

应用程序必须按照 CLDC1.0、MIDP2.0 打包。

1.2.7 应用程序的闪屏（Splash screen）

- 必选：**
- 1、在启动应用程序的同时应启动一个进度条：“正在下载”进度条，用以表示应用初始化的进度
 - 2、在进度条显示之后（应用初始化完成后）应显示闪屏
 - 3、闪屏应只显示一屏：开发商的 LOGO 屏
 - 4、开发商的 LOGO 屏内包含：
 - 应用程序的名称或 LOGO
 - 应用程序的版本
 - 开发商的名称或 LOGO

- 可选：**
- 1、在出现开发商 LOGO 闪屏时可以伴有音效。
 - 2、可以允许用户按任意键跳过闪屏

1.2.8 应用程序的主屏幕（Main screen）

- 在闪屏过后显示的是应用程序的主屏幕或基本屏幕
- 主屏幕至少应包含以下各项：
 - ◆ 开始：启动应用程序（此菜单项名称由 CP 自定义）
 - ◆ 帮助：提供应用介绍，操作指南等。建议图文并茂。
 - ◆ 退出：退出应用程序。
 - ◆ 设置：对应用的可设置项，所有应用必须要支持声音，因此至少要包含声音设置项。
 - ◆ 关于：应用的基本信息
 - 应用的名称
 - 版本号
 - 公司名称
 - 支持电话
 - Email 地址
 - 公司网站（如果有）
- 测试中心将检查所提供的支持电话和 Email 的真实性，如果为虚假号码和地址，则

不予受理。

- 菜单项必须为中文
- 菜单必须为图形化界面

1.2.9 运行过程的等待处理（Process waiting）

很多情况下，应用程序无法做出瞬间反应。如果用户必须等待应用程序完成一个进程，应向用户提供一个可视的反馈，并应提示用户如何取消进程。

- 小于 1 秒：一秒钟是正常思维速度的极限，但用户可能注意到延误，不需要提供反馈。
- 1 秒以上：当用户在等待长时间的进程时，应使用动画等(如，进度条)来显示进度状态，建议给出提示信息（如，需等待的时间）。

对重复按键，除“CLR”键外，予以忽略，因为用户在进程处理时间较长时倾向于重复按键。

1.3 JAVA 用户接口特性

1.3.1 版式（Typography）

1.3.1.1 字体

推荐中文字体表

适用于 LCD 分辨率	Java 应用分辨率	字体名称	字体规格
128×160	128×128	小	11×12
		大	15×16
144×176 132×176	132×148	小	11×12
		大	13×14 或 15×16
176×220	176×184	小	11×12
		中	15×16
		大	19×20
240×320	240×280	小	15×16
		中	19×20
		大	24×24

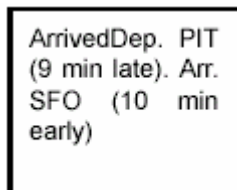
。

1.3.1.2 对齐

为方便用户阅读，文档应左对齐，标题可以左对齐、居中或右对齐。

1.3.1.3 文本换行

为使信息更容易阅读，应该在适当的地方换行，如下：



ArrivedDep. PIT
(9 min late). Arr.
SFO (10 min
early)

未换行的文本



Arrived
Dep. PIT
(9 min late)
Arr. SFO
(10 min early)

换行的文本

1.3.1.4 屏幕布局统一

由于手机终端的屏幕大小的差异，使得应用的布局在每一款终端上会有所不同，所以在提交 JAVA 应用的时候要针对每一款手机来做相应的提交，联通会确定几种手机的屏幕大小，应用可以满足于这几种屏幕的要求，即可以在符合联通规范的所有手机终端上运行。

1.3.2 图标 (Icons)

◆ 图标分类：

- 状态图标：若菜单项有多个状态，可使用图标直观的显示这些状态。
- 图标栏：当向用户提供一系列操作，但又不能占用太多应用程序的屏幕面积时，应使用图标栏。

◆ 图标使用规范：

■ 图标尺寸

应用程序的图标

- 每个应用程序的图标尺寸：16×16。
- 尺寸为 16×16（或 26×26）的图标，实际占用尺寸为 12×12（或 22×22），保留 2 个点阵宽度的透明边界或者图标进行透明化处理。

菜单的图标

- 12×12 (list menu)，图标高度不大于文本高度
- 24×24

■ 图标的格式：

PNG （透明化处理背景颜色 RGB 值=0,0,0)

1.3.3 提示文本（Alerting text）

- 提示文本应为简体中文，避免出现其他语言和中文混合的内容（英文专有名词除外），如：“Loading...请稍候”，应为“下载中...请稍候”。
- 错误信息提示应简明，并提示用户如何退回到前一屏幕。

1.3.4 菜单（Menus）

◆ 菜单用途：

- 操作清单：菜单列出了用户可选的操作，可用不同图标来标识菜单选项，区分不同操作。



- 内容选项：列出一套供用户选择的内容项，不需要为其提供图标。例如：在铃声下载应用程序中，可能有一个菜单列出用户能够下载的全部铃声。



- 应用程序选项：为应用列出供用户选择的选项，可用复选框或者单选按钮进行选择。



◆ 菜单使用规范：

- 菜单标题字数控制在一行显示
- 缺省的情况下，菜单中的每一项占据一行，但是在特殊情况下，允许在同一行中使用多个菜单项
- 菜单中的各项要按照一定的逻辑顺序来排列，如按照类型、时间、字母顺序等。如果没有逻辑顺序，就按优先级排列，将优先级高的排在前面
- 当设计使用菜单的用户界面时，要考虑菜单层次的宽度和深度。宽度指每层菜单中选项个数，建议选项个数以终端一屏显示为上限；深度指结构中的菜单层数，建议 2~3 层。
-

1.3.5 文本输入和导航（Text input and navigation）

文本规范：

- ◆ 若需要特定输入格式，都必须标明输入格式，例如，输入日期：MM/YY
- ◆ 若输入超过一行，文本框应支持多行输入和滚动条。
- ◆ 对用户输入的密码，在输入过程中应用 * 进行掩盖。
- ◆ 支持输入数据的自动记忆功能，如用户名和密码的保存。
- ◆ 对需要用户名和密码登陆的应用，要充分考虑如何让新用户非常方便、简单地完成注册；对用户注册后再次使用时，对于不涉及移动支付等安全性要求非常高的应用，建议用户名和密码默认填好，用户选“登录”键即可。
- ◆ 对用户输入应自动切换到恰当的输入模式

例如，若应用程序要求用户输入电话号码，自动将文本输入模式设定为只能输入数字。

若要求用户输入用户名，由于用户一般会用字母表示，所以自动将文本输入模式设定为字母。
- ◆ 应用程序应限制用户在要求字符数的范围内输入，以避免用户输入错误
- ◆ 对确定的内容进行预置，例如，输入“年”可以预置为“20XX”，只需用户输入后两位数字
- ◆ 尽量减少用户的文本输入
- ◆ 为每一个输入项作直观的提示与描述，不多于 8 个汉字

- 在文本输入的状态下“后退/返回”前一屏幕

在文本输入的状态下，用户可有两种不同办法返回前一屏幕。首选办法是选择软键菜单。在文本输入的软键菜单中加入一个菜单项，其功能类似于菜单中的“后退”或者“取消”功能。

第二种方法是，在文本输入中但无字符输入的状态下，允许“CLR”键起到“后退”或者“返回”前一屏幕的功能。

应该考虑将上述两种方法合并起来使用。

1.3.6 表单

表单是与用户交互的常用界面，以下列出了表单布局中的一般建议：

- 数字应靠右对齐。
- 货币字段的数字应靠右对齐，但货币符号应靠左对齐。
- 标题和输入字段的标题应清楚分开。
- 将输入字段排序和分组是一种有效办法，用户不需要经常前后翻阅。
- 应在视觉上将标题及其输入字段相联系。

1.3.7 关键词（Key words）

在应用中，一致的关键词使用风格能显著提高业务的可用性，一般而言，关键词的首字母应大写，其他字母应小写，除非纯大写组合的词，如：OK 等

1.4 UniJa 终端基本特性

1.4.1 数据传输协议

移动终端支持 HTTP 1.1 协议，支持 TCP/IP 协议，支持 SSL V3。

1.4.2 默认主页地址

联通 Java 的默认 WAP 主页地址为 <http://10.199.17.100/4g/w/>

1.4.3 应用索引

移动终端必须支持用户通过输入 Java 应用程序的索引号，按硬键“联通无限键”，启动 WAP 浏览器自动登录到索引号对应的页面。

1. 索引号格式为最长为 8 位的半角数字字符串。
2. 索引号使用示例



图 2 索引号使用示例

3. 索引号对应的 WAP 页面的 URL 格式如下:

http://10.199.17.100/4g/w?appCode=6xxxxxxx&mc=Search_AppCode

1.4.4 目录管理

移动终端可以有分类管理下载的内容,有能力创建若干个目录并且可以删除某个或全部目录。

1.4.5 系统设置和应用设置

系统设置 (Device Setting) 是指否静音、音量大小、是否振动、背景灯是否开、红外/蓝牙是否打开等终端设备上的设置。应用设置 (MIDlet Setting) 是指 MIDlet 运行时对上述系统资源的控制,用户可以在 MIDlet 运行过程中修改应用设置,但修改仅对当前运行的 MIDlet 有效。两者之间的关系如下表所示:

音量		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF
振动		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF
背景灯		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF
红外/蓝牙		

系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF

音量		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF
振动		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF
背景灯		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF
红外/蓝牙		
系统设置	应用设置	运行时结果
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF

1.4.6 与 R-UIM 卡的关系

通过 OTA 下载的 Java 应用需要和移动终端使用 R-UIM 卡的 IMSI 号绑定在一起实现安全管理。

在 JAM 侦测到 R-UIM 卡更换后，应：

- 1、提示用户“R-UIM 已更换，是否更新用户注册信息？（Y/N）”。如果用户选择“是（Y）”，执行 2.2.6.6 中规定的用户注册流程，然后按下列 2 中规定操作。如果用户选择“否（N）”继续执行下列 2 中规定操作。
- 2、对于非使用当前 R-UIM 卡购买的应用，不能执行：运行/升级/Suspend/Resume 等操作；可以执行：删除/Renew 操作。对于不能执行的操作，当用户换回原用于购买的 UIM 卡后，恢复相应操作权利。

1.4.7 用户注册

在 JAM 第一次启动或侦测到 R-UIM 卡更换后，提示用户“R-UIM 已更换，是否更新用户注册信息？（Y/N）”。用户注册以 JAM 通知移动终端发送 SMS 的方式执行。

OTA 服务器收到注册短信后，更新 IMSI 和 MDN 的对应表，完成用户的注册，然后返回通知短信告知用户注册成功。

如果由于某些原因，用户在服务器端注册未成功，用户下一次访问 OTA 服务器时，服务器将返回提示页面，提示用户手动选取 JAM 中的注册菜单。

1.4.8 Flash ROM

ROM 为移动终端上的静态存储空间，包括对于终端系统平台的存储空间和 Java 应用及相关数据的存储空间。

终端客户端软件需要的存储空间至少为 2.1MB。

移动终端应为 Java 应用及相关数据（RMS）预留出至少 512 KB 的存储空间。

每个 Java 应用占用的存储空间（包括 JAR 文件及其 RMS 存储空间）不能超过 256KB。JAM 应对此进行判断和控制。

1.4.9 RAM

RAM 为终端动态数据的存储空间。运行 UniJa 的终端所需的 RAM 的最低配置为 1.5MB。

1.4.10 LCD

移动终端必需支持彩色屏幕显示。

必须至少支持 12bits 彩色显示；联通建议为 16bits 彩色显示。

屏幕分辨率要求如下：

序号	LCD 物理分辨率	应用可使用分辨率
1	128×160	128×128
2	176×220	176×184

应用可使用分辨率不包含LCD上部的图标状态栏和软键区域。支持增值业务的终端的显示区域每行应至少支持10个中文简体字符的显示。

1.4.11 声音

必须支持的声音编码和格式：

- ♦ Tone Sequence
- ♦ SMF Type 0
- ♦ 13K QCELP

建议支持的声音编码和格式：

- ♦ SMAF Phrase

1.4.12 图像

必须支持的图像格式

- ♦ PNG 1.0
- ♦ ISO/IEC JPEG and JFIF 1.02
- ♦ GIF 89a

1.4.13 摄像头

联通建议支持摄像头。

1.4.14 软件要求

移动终端必须支持 J2ME 技术，通过 JTWI TCK 测试。

移动终端必须拥有 WAP 2.0 Browser。

移动终端必须拥有 JAM。

移动终端必须支持 [29]《中国联通增值业务数字版权管理系统规范 Ver. 1.0》。

移动终端必须支持 CUCL 1.1（注：如果移动终端无摄像头，CUCL 1.1 中有关摄像头部分为可选）。

1.4.15 用户界面与操作

用户可以通过进入“联通无限”选择“神奇宝典”启动 JAM。

用户也可以通过快捷键直接启动 JAM。

1.4.16 测试证书

联通测试证书时效期为 21 天。超过 21 天，就需要重新申请。

中国联通作为唯一的 CA 签发证书。

1.4.17 暂停

只要对应用可以进行暂停，就要求提供暂停功能

1.4.18 软键

- ◆ 目前 UniJa 支持左、中、右三个软键，因此 CP 要尽量充分利用这三个软键。
- ◆ 凡是不显示软键的应用，要求在应用运行当中能够调出主菜单，建议通过右软键调出主菜单。
- ◆ 如果应用使用了如声音、振动等效果，要求在应用运行当中能改变设置

2 开发设计指南

由于手机硬件设备方面的限制及差异，我们在开发游戏的过程中不得不考虑到各方面的因素，以下讨论了一些关键的性能参数，这些参数直接影响游戏的效果。

2.1 分辨率

手机的屏幕大小各不相同，分辨率的高低直接影响造型的大小和表现力。对于美术设计人员，在设计之初就要充分考虑到屏幕的自适应问题，如：在 128×128 分辨率下与 200×200 的分辨率下，如何可以不重新更换图片就可以达到同样的效果。

2.2 色彩数量

目前符合联通 UniJa 规范的手机至少支持 4096 的色彩数量，因此。可能有的手机号称支持 4096 色，可是实际的 4096 色图片放到手机里会发现有的颜色无法区分，而且有的还会出现色偏，所以设计的时候一定要根据实际手机进行图片绘制。

2.3 明亮度

液晶屏幕由于其独特的发光原理，并不能达到传统显示器的亮度，同时也就导致了很多彩丰富的图案在强光下，不能显示出原有的效果，这点在户外使用手机是相当明显，因此设计人员在设计手机游戏图片时一定要考虑到这一点，不能将色彩对比度设置的过于接近。

2.4 显示速度

在手机刷新屏幕的时候，由于运算速度的问题，导致动画显示不能流畅，因此，要尽量采用局部刷新或尽量避免全屏幕动画。

2.5 电力供应

手机的液晶屏幕在高亮度及低亮度显示上耗电量是不同的，例如白色背景要比黑色背景消耗更多的电池电量，因此，在设计游戏时，尽量用低亮度的背景来节省电量。

2.6 执行速度

程序员的经验法则告诉我们：程序往往用 90%的时间去运行 10%的代码。因此，与其努力提高代码的效率，不如找出代码中的瓶颈部分，使之更高效，这样会得到更高的速度。

另一条众所周知的法则是：精细的设计和算法选择比逐行的代码优化更为有效，这条法则不仅对其他程序，而且对 MIDlets,同样有效，随意的设计选择很容易降低 MIDlet 的效率。

Java 程序通常只花很少的时间执行程序代码，大部分时间都用在执行其所调用的库代码。因此，应该了解各种库的性能（特别是各种图形库）。

性能表现指标不仅随手机型号的不同而不同，而且在相同型号不同版本的手机之间也存在差异。制造商往往会在某个型号的产品生命周期内升级软件版本，有时甚至更换硬件元器件。

2.6.1 程序执行速度

2.6.2 量度

寻找程序中性能瓶颈的常用工具是 Jprobe profiler，但是运行于某台手机上的 MIDlet 往往不能运行 profiler，而仿真器的瓶颈与实际手机中的瓶颈有很大差别，因此运行于某台仿真器上的 MIDlet 的场景并不能向我们提示更多的有关信息。

MIDP 的通常方法是在程序中添加一些附加行来自行检测。以下语句可以测出某个调用所费

的时间:

```
Long startTime = System.currentTimeMillis();
DoSomething();
Long timeTaken = System.currentTimeMillis() - startTime;
```

你将得到单位为毫秒的所费时间值, 为了避免测试过程中因垃圾收集而造成的结果波动, 在测试之前, 应先调用 `System.gc()`。要显示测试数据, 可以使用一个特殊的 MIDlet 屏幕, 或者用结果数据覆盖普通屏幕上的显示。

值得注意的是, 必须检查手机系统时钟的精度, 返回值可能精确不到毫秒, 所以请注意检查返回值, 如果它总是十的倍数, 那么保证足够长的测试时间是可以忽略这个问题的。

2.6.3 图形操作

通常, 我们并不关注诸如 `Form` 和 `List` 等 MIDP 高级屏幕类的图形操作执行速度。我们只关注用于动画和游戏等的低级 `Canvas` 类的执行速度。`Canvas` 的图形操作速度随手机型号的不同有很大差异, 因为执行速度不仅取决于基础硬件, 而且也取决于手机自身图形库的效率。在各种支持 MIDP 的手机上, 你每秒钟可以画数以千计的短线, 或数以百计的矩形或小图形。如果仅需要改变屏幕的一小部分, 可以用以下方法请求重绘:

```
Canvas.repaint(int x,int y,int width,int height)
```

这样你就可以指定需要重绘的特定区域。`Paint` 方法将重绘由 `Graphics` 参数划定的矩形区域。要注意的是, 如果所发出的众多重绘请求快于设备的处理速度, 它将合并一些重绘请求, 并使 `paint` 方法的重绘矩形区域扩展到能够覆盖所有需要重绘的区域, 这样的话, 一些非重绘区也可能被重绘。

另一个在大多数手机上都有效的方法是: 如果几次重绘之间的屏幕变化比较少, 就使用脱屏图象。这样就可以在脱屏图象上进行修改, 并用 `paint` 方法将修改后的图象复制到屏幕上。另外, 你可以只复制由 `Graphics` 参数划定的矩形区域。

2.6.4 垃圾收集

避免在堆上产生不必要的“垃圾”对象, 相反, 可以很方便地重用现有对象。

`Immutable` (不可变对象) 是一个特例, 它们的状态自对象创建后就不能改变。由于不可变对象有利于代码的可靠和线程的安全, 因此得到了广泛的使用。

然而, 如果不再需要其初始值, 例如, 如果他们代表的值发生了变化, 此时不可变对象极易变成垃圾对象。这样每一次改变都会创建一个具有新值的不可变对象, 而相应的旧对象则被废弃等待做为垃圾收集。就 MIDlets 来说, 通常使用不可变对象的代价要高于它所能得到的利益, 因此, 最好使用可以赋新值的可重用对象。

最熟悉的例子就是 `java.lang.String`。大多数程序员都记不清多少次使用 `String` 时创建了垃圾对象。经典的例子是字符串的连接。看下面的倒排字符串函数:

```
static String reverse(String s)
{
    String t = "";
    for(int i = s.length()-1;i>=0;--i)
        t +=s.charAt(i);
    return t;
}
```

```
}
```

`t+=s.charAt(i)`;并不改变字符串 `t`,因为 `t` 是不可变的。相反,它每次都创建一个新字符串,复制现存的值并添加新字符。这一方法将不必要地创建 `s.length()`个垃圾对象。这个典型例子说明了不可变对象存在的问题。然而,对于字符串却有一个简单的解决办法:类 `java.lang.StringBuffer` 是 `String` 对应的可变操作类,上面的例子可以更高效地重写为:

```
static String reverse(String s)
{
    StringBuffer t = new StringBuffer(s.length());
    for(int I = s.length()-1;I>=0;--i)
        t.append(s.charAt(i));
    return t.toString();
}
```

综上所述,不可变性是利大于弊的。创建少量垃圾对象不是大问题,只要不是每秒钟创建上千的垃圾对象。甚至 UniJa 就可以每秒钟收集成千的垃圾对象。

2.6.5 多线程

多线程令你的 MIDlets 具有更好的性能,这是因为,当一个线程正等待某种条件时(如等待一个 HTTP 响应,或等待用户输入),另一个线程仍可以工作。

需要记住的是,Java 线程不一定具有抢先权,但可以相互协作。因此,代码不应在“死循环”中等待所需条件,而应该在每次循环中调用 `yield` 或 `wait` 方法,如:

```
try
{
    while(!stopped)
    {
        try
        {
            doSomething();
            synchronized(this)
            {
                wait(500);//会释放当前线程锁定的任何资源,来防止死锁的发生。
            }
        }
    }
}
catch (InterruptedException e)
{
}
```

2.7 网络速度

2.7.1 带宽和延迟时间

单纯的网络速度通常按照带宽和延迟时间来衡量

带宽：某个开放连接中的数据传输速率（通常按每秒钟的比特数来计算）

延长时间：单个数据单元从源到目的地所需的传输时间。

这两项指标都有平均值和浮动差。即使平均值是可接受的，如果浮动差过大，用户也是常常难以接受的。

对于大数据量传输，带宽通常对网络速度的影响是主要的，而对小数据量传输，则延迟时间更为重要。

这两项指标的量度都取决于网络技术和网络连接的特定属性。所以，概括地给出精确数据是不可能的。但可以给出一些概念化数据以帮助设计。

2.7.2 限制 HTTP 的往返传输

由于无线网络中的 HTTP 延迟时间较长，因此应该努力减少 MIDlet 使用的 HTTP 往返传输次数。尽管互联网浏览器可以进行多次 HTTP 往返传输来获得某个网页上不同的框架和图象，但 MIDlet 却应该一次性地获取它所需的全部内容。

如果 MIDlet 要从几个不同的数据源收集数据，一种可选方法是用一个代理 servlet 来收集，这样 MIDlet 仍然只需要发送一次请求。

2.7.3 避免低效协议

SOAP 等基于 XML 的复杂协议可能由于数据大小、解析时间和解析代码大小等高端开销而变得非常低效。如果你正在设计用于某个 MIDP 客户端的基于 XML 的协议，应该使其尽可能的简洁。

如果不需要这些协议的高级功能，则可以用一个简单的自定义协议（例如 `username = fred,password = secret`）来发送同样的消息，这样效率会高很多。

一般来说，如果你使用某个简单自定义协议，那么你就要既写 MIDlet 客户端也写服务器（如 servlet）。一个要注意的问题是版本管理，当升级客户端和服务端时，服务器如使用一个新协议，而对应的 MIDlets 仍然使用旧协议，就会产生一些问题。在客户端和服务端间的开放式通信中包含协议的版本号是明智的。

如果你的 MIDlet 与并非你自己写的或不受你自己控制的服务器通话，或者某台服务器要与除你的 MIDlet 之外的其他客户端通话，则你的 MIDlet 可能不得不用 XML 或 SOAP 与之通话。即使在这种情况下，你也可以使用代理 servlet 来避免这种情况。

代理 servlet 在 SOAP 或 XML 协议与 MIDlet 自定义协议之间实现转换。它根据每个 MIDlet 请求对应地发送一系列 SOAP 或 XML 请求----由于代理 servlet 和服务端之间通过一个快速固定连接，因而这样做的效率很高。

2.8 文件大小

尽量缩小 JAR 文件的尺寸，联通 UniJa 规范规定应用的最大文件大小不应超过 256k，其理由如下：

MIDlet 套件的下载将更快；

用于大规模市场的 MIDP 手机可能只具备不足于应付 MIDlet JAR 文件的有限存储空间。

用户更偏爱小的 MIDlet，因为这样手机就可以一次获得更多的 MIDlet。

2.8.1 小尺寸设计

有两种尺寸比较重要：MIDlet Suite 的 JAR 文件的尺寸以及 MIDlet Suite 安装在手机上所占用的空间尺寸（如果 JAR 文件不是按其打包状况安装的）。后者的尺寸取决于手机的具体实现。但是，在任何情况下 JAR 文件的大小都是所需安装空间的预测标识。

由于 JAR 存档格式具有为每个文件独立设置的头，因此通常情况下最好尽可能使用较少的类。由于这个原因，MIDlet 不象普通的 java 程序那样“面向对象”。特别是：

每个“事物”只有一个类---例如，如果用一个类就可以完成所有事情，就不要把它分成“模型”、“视图”、和“控制器”等类。

限制使用各种接口---接口是一种特殊的类，按照定义，它不提供功能；仅当在同一个发布的 MIDlet 中需要处理多重实现时才使用这些接口。

使用“无名包”---将所有的 MIDlet 类放在同名的包中只会增加 MIDletJAR 文件的尺寸-只为导入库而涉及 package 语句。

考虑使用源代码预处理器，而不用‘static final’常量---每个这样的常量都会占用 JAR 文件的空間。

限制使用静态初始化程序---java 类文件格式并不直接支持静态初始化程序，而是在执行时才作分配；例如，静态地初始化某个字节数组（‘static byte[] data = {(byte)0x02,(byte)0x4A,...}’）在生成的类文件中将导致每个值占用 4 个字节。

2.8.2 库

在一般的软件开发中，开发并使用一些经常用到的库是明智的做法。但是，如果整个库都被包含在 MIDlet 套件中，你可能就要为许多实际上并不需要的功能花费系统开销。

对开发者的开发进度来说，这样做是低效率的，但对 JAR 文件的空間却是更为高效的做法。使用库时，需要考虑是否真正需要库中所有的类。也许可以删除一些类文件。也许需要删除一些不用的方法后对某些类重新编译。

需要记住的是，编写 MIDP 库时应该尽量减少各个类之间的关系，这样，不需要的类就可以真正被安全地删除。遗憾的是，减少类之间相互依赖的典型方法是使用 java 接口，而这又会增大库的尺寸。

2.8.3 保持有限资源

MIDlet 套件通常包含相应的 PNG 位图等资源。应该尽可能使其保持小尺寸、小数量。

我们发现，用不同的位图编辑工具储存 PNG 位图时，位图文件的大小会有很大的不同，它们并不都对尺寸进行优化。可以尝试其中的一些工具，并用储存尺寸最小的工具来保存位图，如使用 PhotoShop7.0 来实现位图的编辑和存储。

2.8.4 考虑合并图象文件

尽量减少类文件有助于降低每个 JAR 文件的大小，同样，尽量减少图象文件对此也很有帮助。一个经常使用的窍门是将许多图象合并到一个文件里。



合并后的图像文件

从文件载入图象后，可以用以下方法绘出每个 frame:

```
g.setClip(x,y,FRAME_WIDTH,FRAME_HEIGHT);
```

```
g.drawImage(fiveMenImage,x-FRAME_WIDTH*frameNumber,y,Graphics.TOP|Graphics.LEFT)
```

这里，frameNumber 的值从 0 到 4；将其按 { 0, 1, 2, 3, 4, 3, 2, 1 } 的顺序进行循环，你可以制造出一幅某人正在行走的动画。

2.9 资源利用

2.9.1 堆内存

移动电话的堆内存不大。

必须确保释放不再需要的屏幕（如 splash 屏幕），以便垃圾收集。不经常使用的屏幕（如帮助，选项等）应该在用到时才创建，并在每次使用后通过垃圾收集而释放其所占空间。这样做就是通过牺牲执行速度而换取了额外的堆内存。

要特别注意内存泄露。当不再使用某个对象，但却在某些地方仍然存在对它的引用时（这样将使其不会被作为垃圾收集），就会发生内存泄露。当不能很快摒弃对某个不再用到的对象的引用时，应将引用设为空。

2.9.2 网络

随着分组交换数据的使用，MIDP 设备正在日益成为主流。所以，它有可能成为各种 HTTP 连接所使用的网络技术。因此你的用户可能正在为你的 Midlet 所形成的网络服务按信息包或按兆字节付费，所以你有责任谨慎使用用户手中的钱。

最小化你所发送的信息包的数量和大小。例如，仅当发生变化时才发送信息包，而不要发送

“什么都没发生”这样的信息。可能的话，避免将你的 MIDlet 设计成向服务器发送一成不变的消息，例如，让 MIDlet 和服务器预测各种状态的变化，仅当状态与预测内容不同时才发送消息。

2.10 响应性能

实际上，用户并不在意你的 MIDlet 有多快，而更在意你的 MIDlet 感觉起来有多快。有很多窍门能让 MIDlet 感觉起来更快，尽管其中的一些方法使实际速度有所下降。

2.10.1 进程提示

用户能很快感觉到 MIDlet 的显示已经停止了变化。此时他们会怀疑 MIDlet 或手机发生了故障。如果进行大型计算或 HTTP 网络访问这类需时较长的工作，最好显示一个动画指示器，如类似标尺的进程指示器。

2.10.2 响应

如果某个 MIDlet 不能迅速反应用户的按键操作，就会被感觉为响应迟钝。就 MIDlet 而言，响应的正常速度是用户对本机应用的体验，它根据手机型号的不同而有所不同。当然，任何超过一秒的操作都被认为是非常漫长的。

要提高响应速度，必须确保事件回调（如 `Canvas.keyPressed` 或 `CommandListener.commandAction`）的快速返回，因为负责重绘屏幕的线程也可能会调用它们。如果要做一个耗时较长的动作，应该由一个独立的线程启动，并设法让用户可以选择退出。

需要记住的是：仅有速度是不够的，用户必须看到工作有所进展。对每一次按键，都应该给用户一个可视的（或可听的）反应。

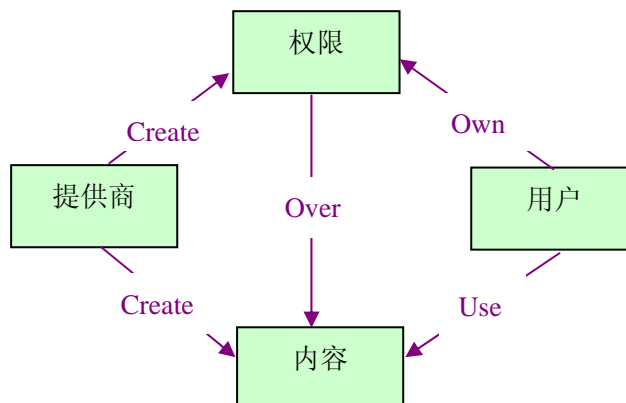
延迟掩盖

用来掩盖延迟的好方法是使用 Splash 屏幕，当 MIDlet 的其它部分还在对自身实施初始化时，可以预先显示这种屏幕。MIDP Alerts(提示)非常适用于 Splash 屏幕，如果你想在 Splash 屏幕超时或被取消时能方便地回调，可以使用 Form 来实现 Splash 屏幕。

2.11 DRM 介绍

2.11.1 概述

数字版权管理（DRM：Digital Rights Management）为内容提供商和运营商控制用户对下载内容的使用提供了管理手段，能够有效的防止盗版，维护内容提供商的利益。



DRM允许内容提供商（CP）制定内容的使用规则（称之为权限）。一个内容可以对应多个不同的权限，相应地，这些权限的价格也有所不同。内容提供商可以提供免费“预览”的内容，也可以提供收费的具有使用权的内容。使用DRM机制实质上是销售内容的权限而不是去卖这些对象本身。

有了 DRM，铃声、图片、Java 应用软件、流媒体和其他内容就能够安全地在无线网络上传递，从而维护了版权所有者的权益。内容开发商能够更有效地利用传播渠道向用户展示和出售他们所开发的内容，更愿意提供自己拥有的内容；用户也可以非常方便地获得合法的数字内容。

2.11.2 权限定义

权限描述了内容的使用规则，包括许可和约束两个部分。

许可即允许做什么。对内容下载服务器上的每个内容，根据其类型的不同，有不同的使用许可：

1. Play 许可-----即允许进行 play 操作，如播放音频和视频。
2. Display 许可-----即允许进行 Display 操作，如显示图片和文本内容。
3. Execute 许可-----即允许进行 Execute 操作，如运行游戏或应用程序。
4. Print 许可-----即允许进行 Print 操作，如打印图片和文本。

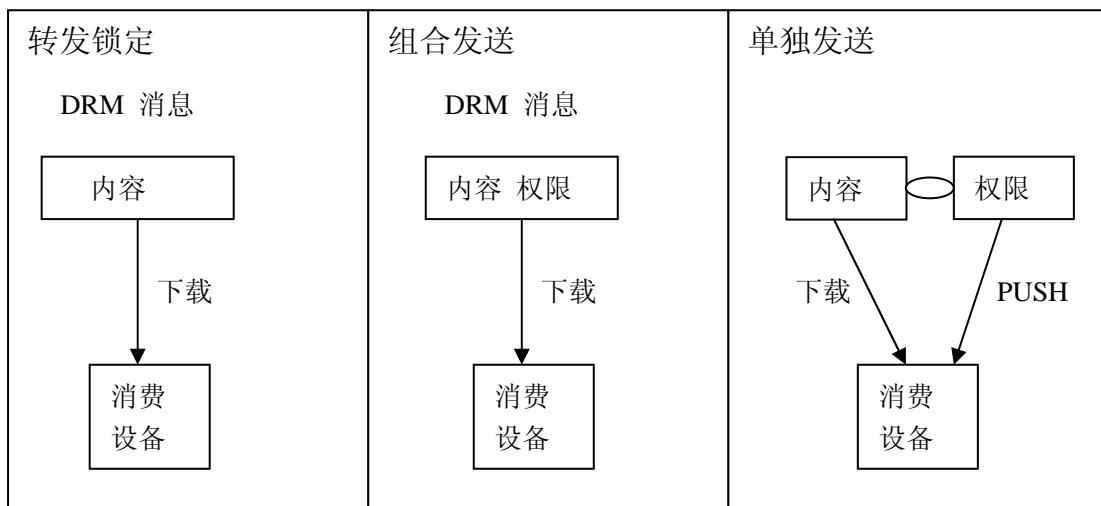
对下载的内容，除了上面的许可外，不允许用户对内容进行修改，也不允许用户转发内容。许可也可以组合，即一个内容可以有多个许可，比如图片既可以 Display 也可以 Print。许可。

约束是在许可之上的限制。对于每个选择的许可，内容提供商可以用约束来限定可下载的内容的使用。每个许可可以有一个或多个下列约束：

1. 计次(count)：规定用户可以使用该内容的次数。
2. 使用时间(Interval)：规定从用户第一次使用该内容开始，用户可以使用该内容的时间，如三个月。
3. 开始时间/结束时间 (Datetime)：规定用户可以使用该内容的起止时间，如只能在 2003 年 9 月 1 日到 2003 年 12 月 31 日之间使用。
4. 永久使用权限：用户一旦购买，即可永久使用该内容，但不能转发该内容。

2.11.3 DRM 方法

目前联通提供三种 DRM 方法：



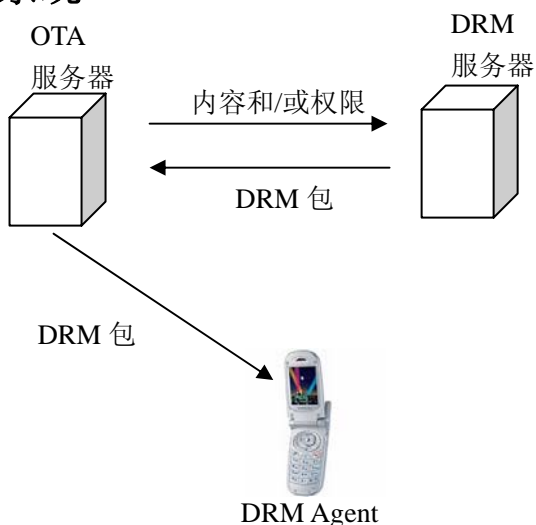
转发锁定适用于传送不许转发给他人的内容。它允许永久使用下载内容、但不允许转发内容。内容本身隐含在传送到终端上的 DRM 消息中。

组合传送允许为内容设置使用规则。这一方法是对转发锁定方法的扩充，它在 DRM 消息中添加了一个权限信息，只有权限有效时才能使用内容。此方法还可以支持预览功能。

单独传送保护高价值内容并且支持超级分发功能。它允许设备转发内容，但不能转发权限。这一方法采用不同的信道传送内容和权限，因此比组合传送方法更安全。利用对称加密方法将内容加密成 DRM 内容格式 (DCF)，在权限中保存了内容加密密钥 (CEK)，这样，设备就可以使用此密钥进行解密。

对于 JAVA 应用，目前只支持转发锁定和组合传送两种方法。

2.11.4 联通 DRM 系统



联通 DRM 系统由 DRM Server 和手机上的 DRM Agent 组成。

DRM Server 将下载内容和其对应的权限打包后供用户下载。

用户下载了带 DRM 保护的内容后，手机上的 DRM Agent 负责解析和管理权限，并通过权限控制下载内容的使用。

因此，内容提供商提供内容的时候，如果需要对该内容进行 DRM 保护，则需要一并提交相应的使用规则和计费方式，以便生成权限、销售权限。

用户下载了带 DRM 保护的内容后，只有在权限有效期内才能使用内容，同时不能转发该内容。

为了方便内容提供商介绍和推广所开发的内容，联通 DRM 系统提供用户预览或试用内容的机会。同时，联通 DRM 系统还提供内容的挂起和恢复以及权限的重续功能。

当用户希望暂时中止已下载内容的使用，从而可以节省空间时，可以选择挂起一个内容。用户可以通过该功能将已经下载的内容删除，释放所占空间，但不释放其权限信息。当用户希望重新使用该内容的时候，可以使用恢复功能，重新下载该内容，从而在剩余的权限期间内继续使用，而不必购买新的权限。

当某个内容的权限已经使用完，用户仍然想继续使用时可以选择重续权限。用户不用重新下载内容而仅仅下载使用权限就可。这样节省了用户的开销和时间，从而更好地方便业务的开展。

2.12 Digital Signature 介绍

MIDP2.0 规范定义了一个安全模型，这个安全模型让 MIDlets 有权访问安全性敏感的 API，联通将通过数字签名来保证 CP 提交的应用可以访问到被授予访问权限的 API。

CP 在开发完应用后，应从联通申请一测试签名用于测试目的，在自测通过以后，提交到联通，联通将用数字证书来对应用做最终的数字签名，签名的应用被分配到相应的保护域，保护域定义了一组许可可以授予该域里的 MIDlet Suite。域是和 ME Keystore (mdpipinstalldir/appdb/_main.ks) 里的公钥相关的，当用 ME Keystool 向 ME Keystore 输入公钥的时候，和这个公钥相关的域或者被指定或者用默认的域 untrusted。

当用户安装已做数字签名 MIDlet suites 时，由 MIDP RI 自动的把 MIDlet Suite 分配到一个域，如果 MIDP RI 没有找到签名的公钥，它就不安装它。

为一个 MIDlet Suite 做签名包括向 JAD 文件里增加证书和为 Jar 文件创建数字签名并把它增加到 JAD 文件里。

当用户安装 MIDlet suite 的时候，MIDP RI 必须确定这个 MIDlet suite 是否是可信任的。通过查找和验证 JAR 文件的数字签名和支持这个签名的证书来做决定。如果 JAR 文件被签名，MIDP RI 希望这个签名是基于 X.509 PKI 的，这是为了 MIDP RI 能验证这个签名，并信任这个 MIDlet suite。。

如果签名的 MIDlet suite 是可信任的，它就被安装并分配到一个域里，否则它就不被安装。

MIDlet suite 运行的时候，MIDP RI 必须能确定它有什么许可。当 MIDlet suite 调用被保护的 API 的时候，被保护的 API 用 com.sun.midp.midlet.Scheduler 类获得当前运行的 MIDlet 的名字，用 com.sun.midp.security.SecurityToken 类查看是否 MIDlet 有权运行这个方法。

MIDP2.0 规范定义了一个安全模型，这个安全模型让 trusted MIDlets 有权访问安全性敏感的 API。

默认状态下，MIDlet suites 都被视为 untrusted，并被绑定到 untrusted domains，从而不能访问任何的敏感性功能。trusted 和 untrusted MIDlets 的区分是通过证书、数字签名来实现的。

经过签名的应用下载到手机之后，经验证证书有效且签名有效之后，将与设备中的某一保护域绑定。随后，设备依据具体情况（如 JAD 文件中的某些属性、JAR Manifest 中的某些属性等）和该保护域安全策略所规定的 permissions，对该应用可最终使用的 APIs 做授权。

上述过程可参见下图。

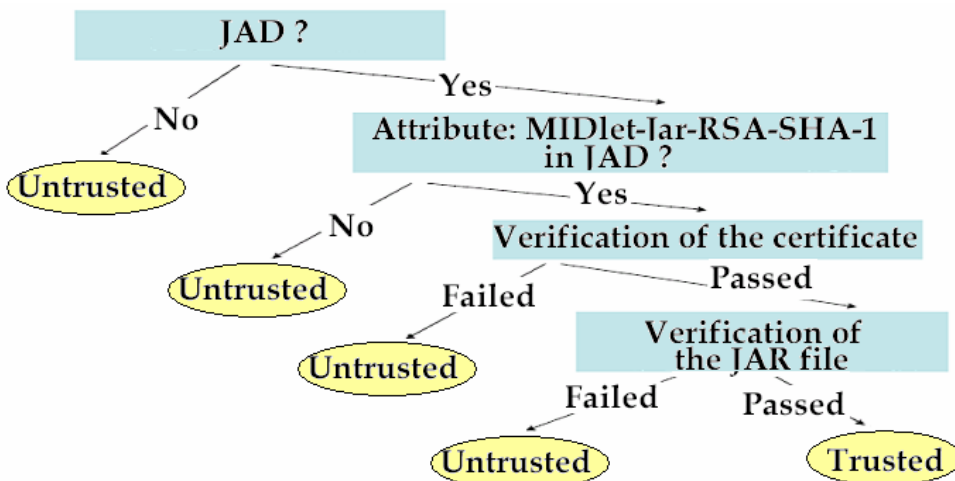


图 Trusted vs. Untrusted MIDlet Suites

（1）CP/SP 开发的 trusted 应用提交、测试及预备下载流程

在 CP/SP 向中国联通提交应用之前，必须拥有一个帐户。OTA 服务器为 CP/SP 建立帐户时，将为该 CP/SP 生成一对公/私钥对，并向 CA 发出证书请求，CA 创建好该 CP/SP 的证书后，发送回服务器。

在 UniJa 中，中国联通将作为唯一的 CA 为 CP/SP 签发证书。

CP/SP 的应用必须联通测试，测试时使用的是联通测试证书。该证书有时效限制，有限期为 30 天，超过期限需重新申请。

测试通过之后，服务器将用该 CP/SP 的私钥为该 CP/SP 提交的 JAR 文件签名，然后服务器将签名和该 CP/SP 的证书插入 JAD 文件的相应属性。这时，CP/SP 的应用才可供用户下载。

（2）CP/SP 的受信等级

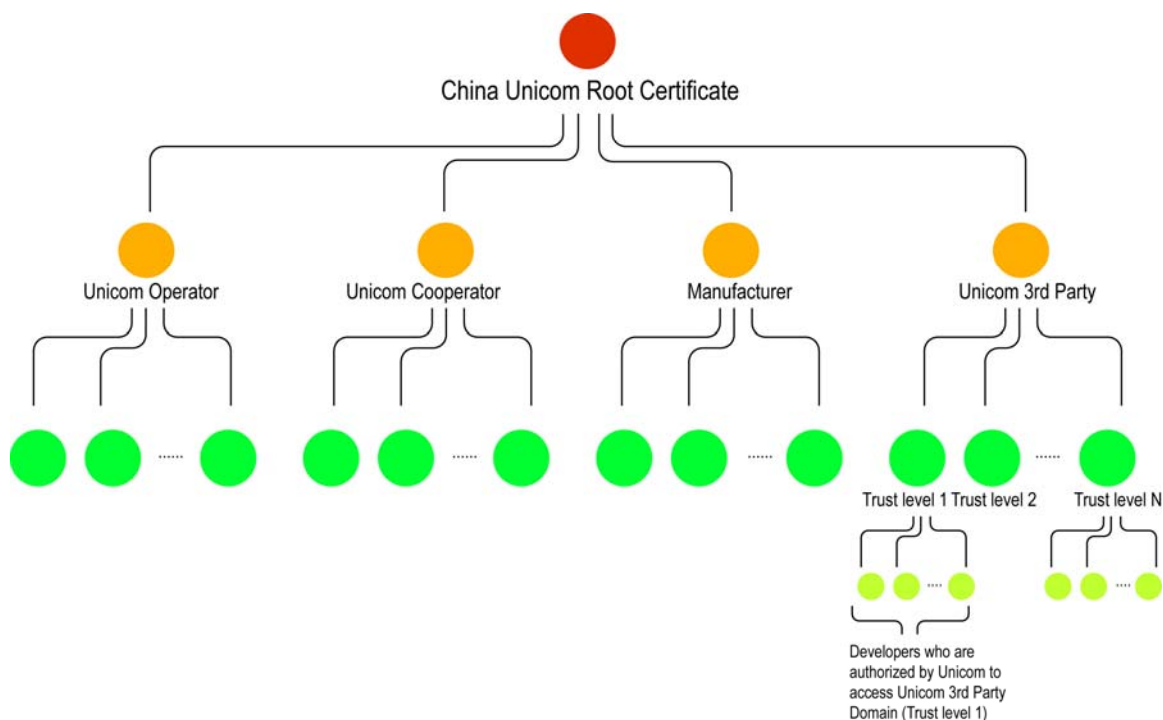


图 中国联通设备安全策略（Device Trust/Security Policy）

在 OTA 服务器为 CP/SP 建立帐户的同时，也为 CP/SP 分配了一个受信等级，该等级对应于联通设备安全策略的某一个保护域，也就是说，CP/SP 开发应用时可以使用的 APIs，包括可以直接访问的 APIs（Allowed permissions）和需得到用户明确授权后才可访问的 APIs（User permissions），已由该保护域的安全策略决定。

CP/SP 必须根据自己的受信等级来开发应用。

然后 OTA 服务器根据 CP/SP 的受信等级在该 CP/SP 的应用通过测试之后，为该 CP/SP 的应用在 JAD 文件中插入对应于该 CP/SP 受信等级的证书。

2.13 声音类型

在手机的铃声关闭的情况下，运行的应用的声音应同时关闭。

支持的声音编码和格式：

- ♦ Tone Sequence
- ♦ General MIDI
- ♦ 13K QCELP

可能支持的声音编码和格式，这需要根据每款手机来确定：

- ♦ 8KHz 8-bit PCM Wav
- ♦ SMAF

2.14 操作键

2.14.1 数字和符号键

数字键对应于移动终端上的数字键 (0-9)，符号键对应于移动终端上的“*”和“#”。

2.14.2 光标方向键

光标方向键对应于移动终端上的方向键(↑ ↓ ← →)。
它们的基本功能是移动光标和滚动屏幕。

2.14.3 上下翻页键

上/下翻页键对应于移动终端上的上/下翻页键。
它们的基本功能是以屏为单位移动光标和滚动屏幕。

2.14.4 清除键

UniJa MIDlet 无需支持终端上的清除键。

2.14.5 终止键

终止功能键对应于移动终端上的终止键。
此键由手机操作系统处理，应用无需支持此键。

2.14.6 管理键（快捷键）

移动终端必须支持管理键。待机屏幕下，短按硬键“管理键”，启动 JAM，显示 JAM 菜单；
长按硬键“管理键”，启动 BROWSER，连接到系统配置的 URL 所指向的 OTA 服务器（缺省配置为联通 OTA 服务器）。

2.14.7 索引键

移动终端必须支持索引键。待机屏幕下，用户输入 JAVA 应用程序的索引号，按硬键“索引键”，终端自动登陆到索引号对应的 JAVA 应用程序的页面。

2.14.8 功能软键

功能软键对应于移动终端上的三个功能键：F1, F2, F3。

当按下一个功能键时，将执行当前屏幕上对应功能软键所指示的功能。

2.14.9 按键事件和 Game Actions

UniJa 应用应该调用 `Canvas.getGameAction` 方法来决定某个给定的 `Key Code` 对应的动作 (Game Action, or Action Event)。建议 UniJa 应用的逻辑建立在此方法返回值的基础上，这样的应用移植性好，并且可以运行在键盘设计不同的 UniJa 手机上。