

i51 Application Developing Kit Reference Manual

Version 1.0.2

Since:2011-09-21

Update:2012-03-19

Editor:Jelo

目 录

第1章 新手上路

[1.1 关于“i51”](#)

[1.2 编译环境搭建](#)

[1.3 套件的概念](#)

第2章 工程实践

[2.1 创建应用程序](#)

[2.2 添加资源](#)

[2.3 编译应用程序](#)

[2.4 发布应用程序](#)

[2.5 创建应用套件](#)

[2.6 发布应用套件](#)

[2.7 运营发布](#)

附录

[1.1 工程模式](#)

[1.2 常用工具的使用](#)

第1章 新手上路

1.1 关于“i51”

“i51”是在底层操作系统基础上搭建起来的一个中间件平台，它为应用层提供了统一的、系统无关的编程接口，它的跨平台性可以让应用团队无需关心操作系统的差异，较好地控制了各个环节的复杂度，因此“i51”应用程序可以运行在所有具有“i51”中间件的设备环境中。

i51 体系结构其架构由“DC 动态层”、“SC 静态层”构成，动态层中的“对象”具备高度灵活特性，应用程序的发布不受操作系统束缚，并且相关编程框架为应用开发提供了统一且平台无关的编程接口。而，静态层与动态层的特点正相反，它的作用是屏蔽操作系统特征，并为动态层提供最基本的运行时机制，因为这一部分较少变化，所以它的实现通常作为内置代码嵌入到设备的ROM之中。

§ Applications

i51 体系的动态层只有一种应用程序，它的实现受“i51 编程框架”约束。应用间处于平级关系，所谓平级是指应用间相互独立、不存在依赖关系。另外，应用程序之间的交互基于静态层提供的“消息机制”完成，整个交互过程受

静态层监管。

§ Dynamic Library

“动态库”是操作系统中非常核心的技术，在 i51 中同样将其作为了体系结构的核心特性，它的价值主要体现在应用层，基于动态库，应用程序生命周期的三个阶段：开发、部署、维护都是完全可重构的，例如按照业务逻辑将应用程序划分多个动态库，当业务发生变化时只需要替换相关动态库即可，而在传统的静态库做法中，一旦业务逻辑发生了变化整个应用必须要重新经过“开发”、“部署”、“维护”三个阶段方可完成业务更换。

§ Static Applications

“静态应用”简称 SAP 是位于静态层的一组应用程序，因为它跟 i51 体系的具体实现有关，因此其功能特征在体系结构中并不作限制，开发者可根据不同需求定义这组应用的特征。

§ Kernel

“内核”负责应用管理、进程管理、资源管理、进程通信等工作。严格来说它也是一个“静态应用”，区别在于它与业务逻辑无关可作为一个体系结构中的标准化部件，如果将其泛化将很难保证体系结构在机制层面的完整，因此它的功能特征必须标准化，开发者需要以此为依据严格实施。

§ SL-API

全称是“Service Logical API”，一组与业务逻辑有关的接口集合，该接口集同时对静态层以及动态层开放，但是其功能特征在体系中同样不做任何约束，开发者可以根据不同需求实现。

§Adapter

“适配器”为上层屏蔽了操作系统特征，提高架构整体的维护性，其接口定义、功能特征作为体系标准化内容实施。

※运行时性能

基于内存快照技术，使应用间可无缝切换，并支持多应用同时执行。

基于动态库技术，一个应用可以有多个动态库组成，而动态库可以由应用任意装卸，简化了应用间的关系。

※维护性能

所有业务逻辑屏蔽在动态层，由开发者通过应用程序或动态库实现。

动态库基于适配器实现，使应用程序开发框架具有平台无关性。

1.2 编译环境搭建

安装“Microsoft Visual Studio 2005”、安装“ADS 1.2”或更高。

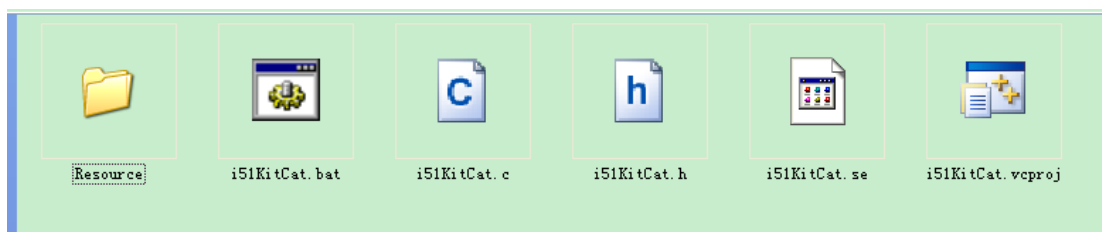
1.3 套件的概念

套件实际上就是上文提到的“动态库”，它的基本特征是“任意一个套件都可以在运行时装卸”。套件在“i51”中有两种具体的定义，分别是：“应用程序”、“应用套件”，应用程序是为了实现某种业务的套件，例如游戏、工具等等。而“应用套件”的功能粒度要小很多，如果说应用程序是“业务级”的，那“应用套件”则是接口级的，它为了应用提供了各类功能接口，应用套件由应用程序加载并调用。

第2章 工程实践

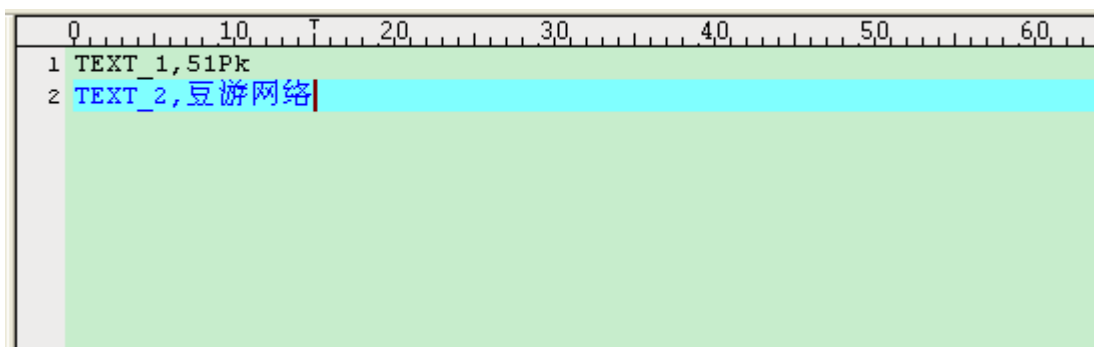
2.1 创建应用程序

双击位于“iSDKi51T00L”目录中的“CreateApp.exe”，输入名称 比如输入 i51KitCat,创建成功则会在上级目录 i51APP 下生成 i51KitCat 应用文件夹，目录下有需要的文件。如图：



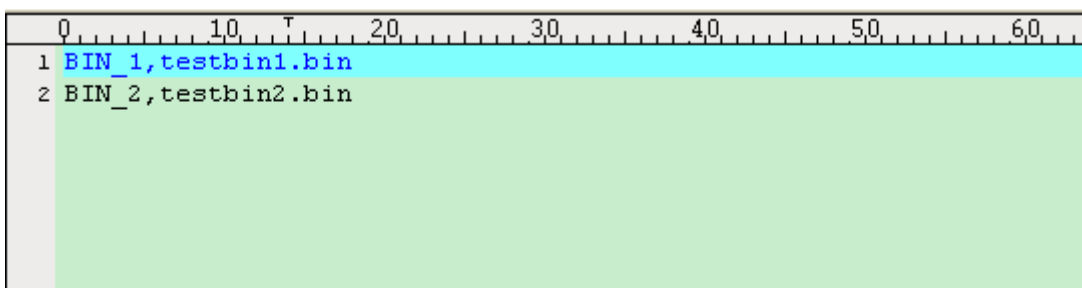
2.2 添加资源

在生成需要的文件之前，需要先将资源准备好，在应用程序或应用套件文件夹下有 Resource 目录，目录下有 TEXT,BIN 两个文件夹，里面分别存放字符资源和其他资源。TEXT\Text.csv 里字符资源书写格式如图：



一行代表一组资源，第一列为 id,第二列为具体数据。(注意 id 不能相同，程序将会通过 id 查找到具体的数据)。

BIN\bin.csv 其他资源书写格式如下：



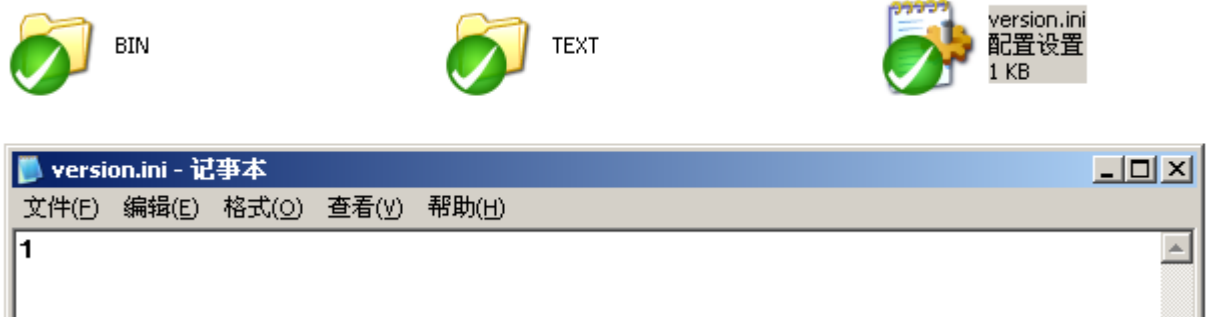
每行代表一组资源，第一列同样是 id,第二列为数据的文件名，文件放在 bin.csv 同级目录下，如图：



资源准备好了，就可以生成文件了双击 MakeRes.exe，输入应用或者是动态库的名称回车（如输入 i51KitCat），然后选择字符串资源编码的方式，0 为 gb2312,1 为 unicode。操作成功后在 i51KitCat 文件夹里面生成 i51KitCat.re 及 i51KitCatRes.h 文件供应用或动态库使用。I51KitCatRes.h 中是一些枚举用来放字符串资源 id 及其他资源 id，调用动态库 i51KitStd 接口即可获得需要的资

源数据。

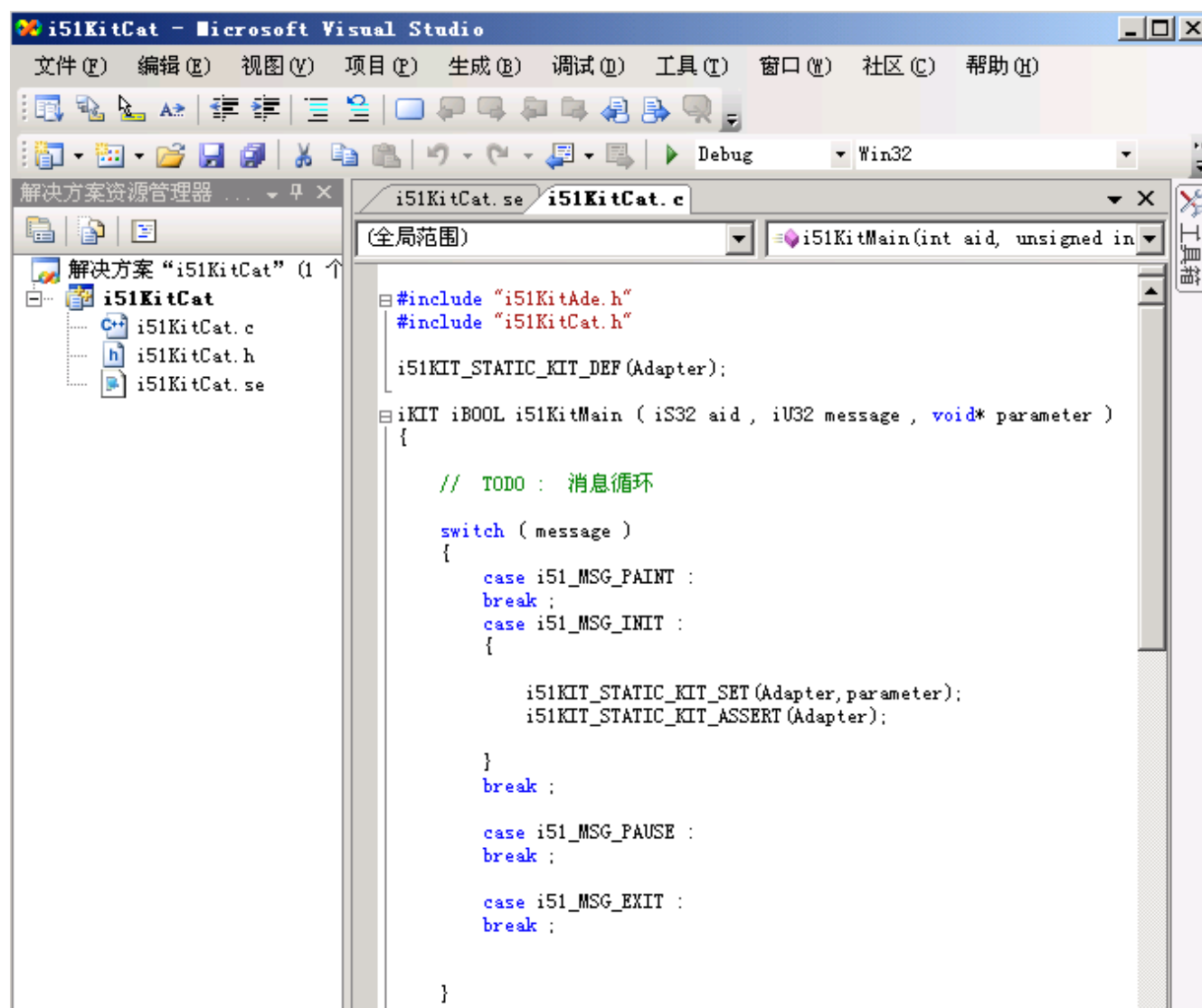
资源的版本



应用需要对自己的资源分版本维护，版本的定义就在应用资源目录下的“version.ini”中，打开文件键入阿拉伯数字即可。

2.3 编译应用程序

同样以“i51KitCat”为例，在“iSDK\i51APP\”目录下找到相应的工程目录，打开“*.vcproj”工程文件，如下图：



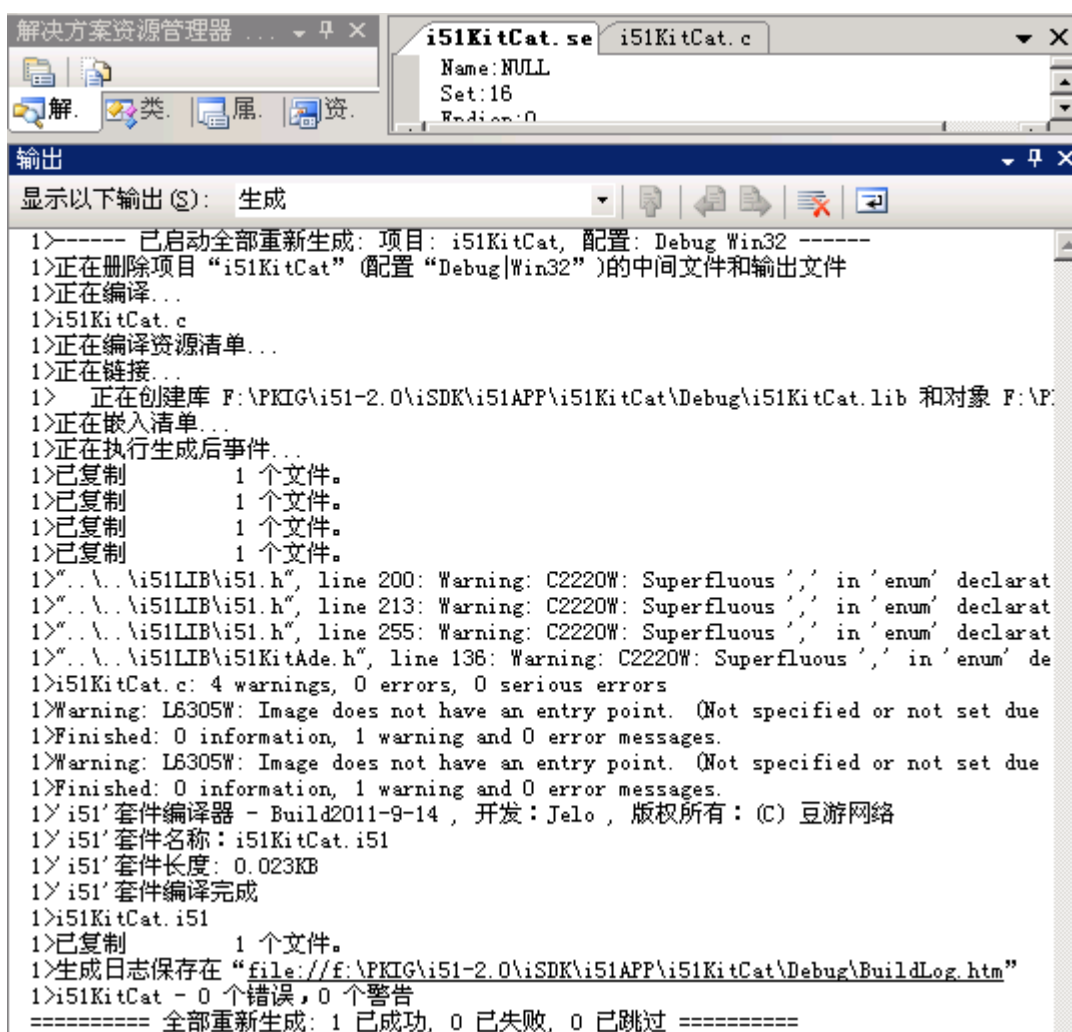
一个“i51”工程至少有 3 个文件，如上图所示：i51KitCat.c、i51KitCat.h、i51KitCat.se。“*.se”为应用程序属性描述文件，需要修改几项信息如下：



- “Name” 为应用程序名称，开发者可以自定义。
- “Set” 为指令集，请保持默认值。
- “Endian” 为大小端，“0” 小端。“1” 大端，开发者可以自定义。
- “Type” 为套件类型，“0” 为 i51，“1” 为 Kit，请保持默认值。
- “Version” 为应用程序版本，最大值 7.31.255，开发者可以自定义。
- “Screen” 为应用程序分辨率，在调用 CreateApp.exe 时填写，开发者可以自定义。
- “Kernel” 所依赖的内核版本，请保持默认值。
- “Adapter” 所依赖的适配器版本，请保持默认值。
- “Weight” 权限，请保持默认值。
- “Resources” 依赖的外部资源，开发者可以自定义。
- “Appkits” 依赖的外部套件，开发者可以自定义。

- “Syskits” 依赖的系统套件，开发者可以自定义。
- “Files” 源代码文件，开发者可以自定义。
- “Language” 开发语言，可填写“C”或“C++”，开发者可以自定义。
- “Compiler” 编译器，可填写“ADS”或“RVCT”，开发者可以自定义。

编译该工程如下图：



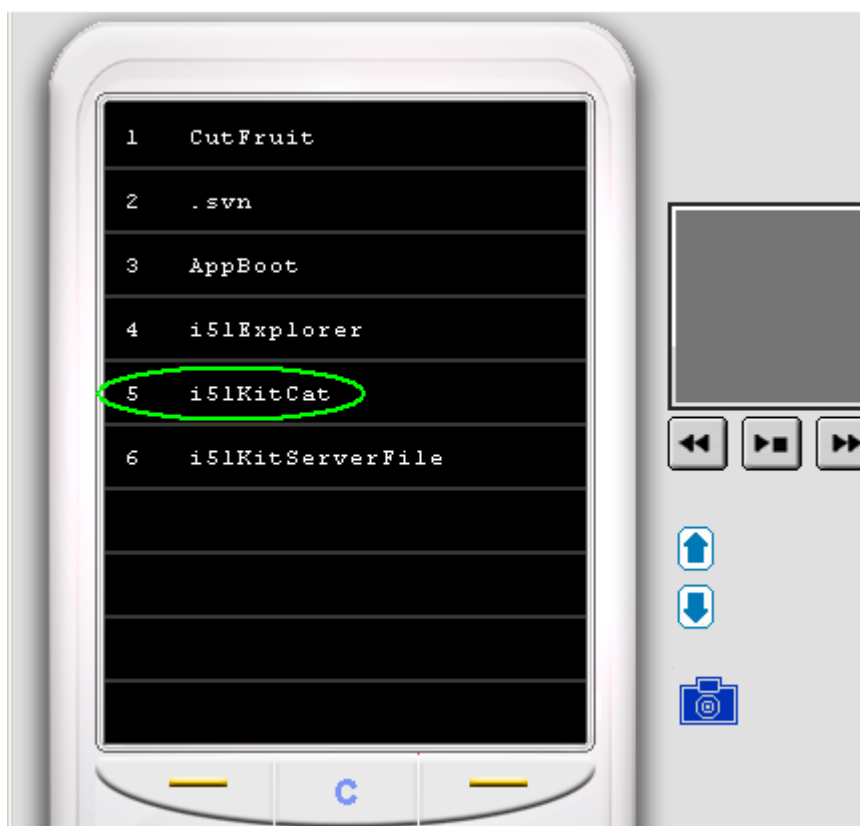
The screenshot shows the i51 Application Developing Kit IDE. The top window is titled "i51KitCat.se" and "i51KitCat.c", showing project settings: Name: NULL, Set: 16, Edition: 0. The bottom window is titled "输出" (Output) and shows the compilation log for the project "i51KitCat" in "Debug Win32" configuration. The log details the steps from deleting intermediate files to linking and generating the final output files "i51KitCat.lib" and "i51KitCat.i51". It also lists several warnings from the compiler (C2220W) and the linker (L6305W) regarding superfluous characters and missing entry points. The log concludes with the successful generation of the project files.

```
1>----- 已启动全部重新生成: 项目: i51KitCat, 配置: Debug Win32 -----
1>正在删除项目“i51KitCat”(配置“Debug|Win32”)的中间文件和输出文件
1>正在编译...
1>i51KitCat.c
1>正在编译资源清单...
1>正在链接...
1> 正在创建库 F:\PKIG\i51-2.0\iSDK\i51APP\i51KitCat\Debug\i51KitCat.lib 和对象 F:\P
1>正在嵌入清单...
1>正在执行生成后事件...
1>已复制      1 个文件。
1>已复制      1 个文件。
1>已复制      1 个文件。
1>已复制      1 个文件。
1>“...\\i51LIB\i51.h”, line 200: Warning: C2220W: Superfluous ‘,’ in ‘enum’ declarat
1>“...\\i51LIB\i51.h”, line 213: Warning: C2220W: Superfluous ‘,’ in ‘enum’ declarat
1>“...\\i51LIB\i51.h”, line 255: Warning: C2220W: Superfluous ‘,’ in ‘enum’ declarat
1>“...\\i51LIB\i51KitAde.h”, line 136: Warning: C2220W: Superfluous ‘,’ in ‘enum’ de
1>i51KitCat.c: 4 warnings, 0 errors, 0 serious errors
1>Warning: L6305W: Image does not have an entry point. (Not specified or not set due
1>Finished: 0 information, 1 warning and 0 error messages.
1>Warning: L6305W: Image does not have an entry point. (Not specified or not set due
1>Finished: 0 information, 1 warning and 0 error messages.
1>i51' 套件编译器 - Build2011-9-14 , 开发: Jelo , 版权所有: (C) 豆游网络
1>i51' 套件名称: i51KitCat.i51
1>i51' 套件长度: 0.023KB
1>i51' 套件编译完成
1>i51KitCat.i51
1>已复制      1 个文件。
1>生成日志保存在“file:///f:/PKIG/i51-2.0/iSDK/i51APP/i51KitCat/Debug/BuildLog.htm”
1>i51KitCat - 0 个错误, 0 个警告
===== 全部重新生成: 1 已成功, 0 已失败, 0 已跳过 =====
```

集成开发环境最终会输出“i51KitCat.dll”、“i51KitCat.i51”两个文件，分别用于模拟器跟真机，开发者不需要关注他们的生成细节。

2.4 应用程序的运行

先看一下模拟器版本，应用程序的运行很简单，完全不需要人为参与，运行模拟器即可运行该应用程序。在模拟器进入“主菜单”，然后进入“娱乐与游戏”然后点击“51PK 游戏中心”，在列表中选择“i51KitCat”，点击运行，如下图：



关于真机运行，将 iSDK\i51EMU\WIN32FS\DRIVE_E\下的 i51 目录整体复制到手机 T 卡即可。

2.5 创建应用套件

应用套件就是那些文件后缀为“*.kit”的，它是一些通用的接口库，为上层应用提供了通用的编程接口，如图形图像库。对于开发者来说可以自己创建“应用套件”为己所用。

双击位于“iSDK\i51T00L”目录中的“CreateKit.exe”，输入名称 比如输入 i51KitTest, 创建成功则会在上级目录 i51APP 下生成 i51KitTest 应用文件夹，目录下有需要的文件。其它与“创建应用程序”无异，在此不一一赘述。

◇ 注意：“*.kit”与“*.i51”有以下区别：

1. “*.KIT”接收不到“i51_MSG_PAINT”消息，“*.i51”可以。
2. “*.KIT”接收不到“按键”、“触屏”消息，“*.i51”可以。

2.6 发布应用套件

与“发布应用程序”无异。

2.7 运营发布

所谓的运营发布是指产品通过了完整的质量测试，最终可以面向用户发布。对于“i51”应用开发者来说，只需要将应用目录下对应的“*.package”文件提交给相关人员即可。

附录

1.1 工程模式

- ◆ 启动自动化测试程序，在拨号界面输入 “*#20111221*#”。
- ◆ 启动应用浏览器，在拨号界面输入 “*#51*#”。

1.2 常用工具的使用

- ◆ 查看 “*.kit” 或 “*.i51” 的依赖项、版本信息等等。

在 iSDK\i51T00L” 中找到 Kitreader.exe，以 “i51KitG2.kit” 为例，在命令行中键入 “Kitreader i51KitG2.kit”，显示如下：

```
F:\PKIG\i51-2.0\iSDK\i51T00L>kitreader i51kitg2.kit
Name:NULL
Endian:Little
Version:0.0.8
OBJ Version:1

KEL Ref Version:0.0.2
ADE Ref Version:0.0.2

Kits & Re Ref Total:1
Kit Ref:' i51KIT\PKiGAPI\i51KitStd.kit' , Ver: 0.2.1
```


◆ 将 “*.i51” 转换为 C 语言数组形式。

在 “iSDK\i51T00L” 中找到 Makesap.exe，以 “i51Logger” 为例：

```
F:\PKIG\i51-2.0\iSDK\i51T00L>makesap
请输入SAP名称:
i51Logger
转换完成
```

完成后将在 “i51KIT\i51Logger” 目录下，生成以下文件：



```
i51Sapi51Logger.c - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

#include "i51.h"
#ifndef i51SIMULATOR
const unsigned int i51Sapi51LoggerSize = 4153;
const unsigned char i51Sapi51Logger[4153] = {
    0x00, 0x88, 0x5C, 0x47, 0x69, 0x35, 0x31,
    0x55, 0x4C, 0x4C, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0xFF, 0x01, 0x00, 0x00, 0x00, 0x10, 0x00,
    0x00, 0x80, 0x00, 0x80, 0x00, 0xA0, 0x00,
    0x00, 0x00, 0x00, 0x54, 0x0F, 0x00, 0x00,
    0x74, 0x09, 0x00, 0x00, 0x0C, 0x00, 0x00,
    0x00, 0x00, 0x00, 0xEC, 0x0F, 0x00, 0x00,
    0x03, 0x00, 0x00, 0x00, 0x74, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x8C, 0x00, 0x00, 0x00,
    0x01, 0x00, 0x00, 0x00, 0x8C, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x0C, 0x03, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0xC0, 0x03, 0x00,
    0x00, 0x00, 0x00, 0x14, 0x05, 0x00, 0x00,
```