



Advanced Card Systems Ltd.
Card & Reader Technologies

ACS 蓝牙库更新指南 [iOS 及 Android]

用户指南 V1.01



目录

1.0.	简介.....	3
1.1.	ACS 智能卡 I/O Android 库	3
1.1.1.	Android 库的安装.....	4
1.2.	ACS 智能卡 I/O iOS 框架	8
1.2.1.	iOS 库的安装	9



1.0. 简介

本用户指南介绍如何在客户创建的应用程序上更新 ACS 蓝牙库。下面是更新蓝牙库的常规步骤。需要注意的是，更新库时可能出现不兼容问题，需要对应用程序做进一步的修改。

1.1. ACS 智能卡 I/O Android 库

“acssmcio”库主要提供与 ACS 蓝牙读写器进行通信的类和接口。该库基于 JSR 268 [1]定义的 Java 智能卡 I/O API 中的服务提供商接口（TerminalFactorySpi）。

Java 智能卡 I/O API 定义了通过 ISO/IEC 7816-4 APDU 与智能卡进行通信的 Java API。该 API 允许 Java 应用程序与智能卡上运行的应用程序进行交互，实现卡片数据的存储和检索等功能。

“smartcardio”库从 OpenJDK 中导入 Java 包：

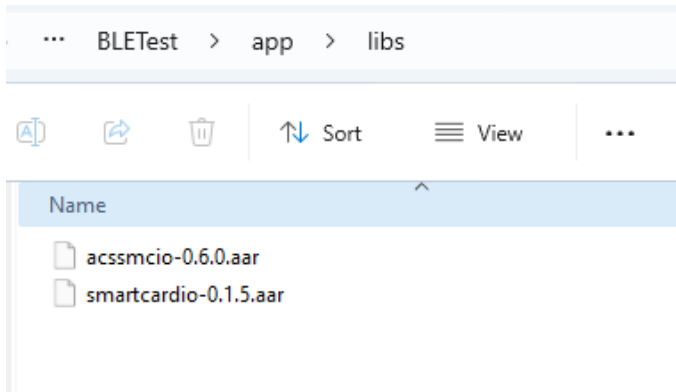
- javax.smartcardio
- sun.net.www
- sun.nio.cs
- sun.security.action
- sun.security.jca
- sun.security.util

[1] <https://jcp.org/en/jsr/detail?id=268>

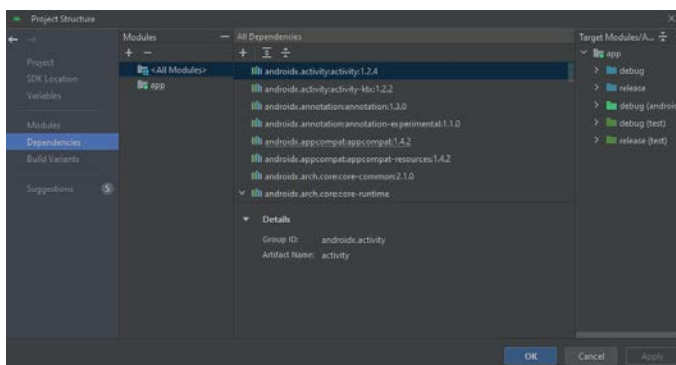
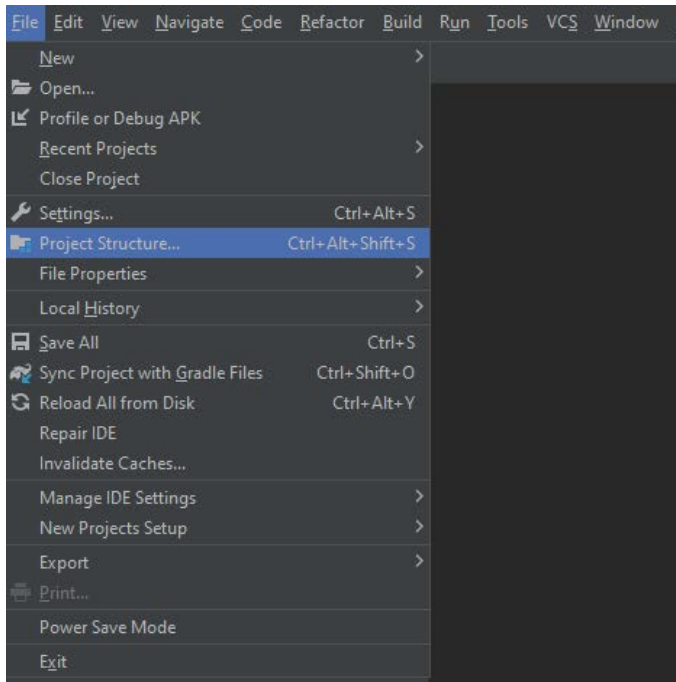
1.1.1. Android 库的安装

1) 下载更新后的 Android 库并解压

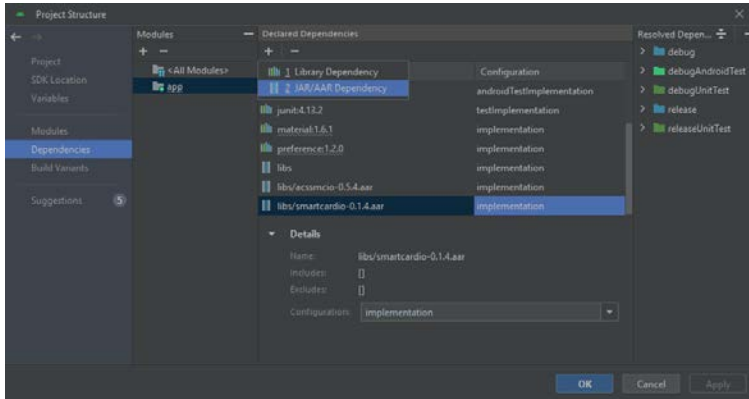
找到解压后的类库文件夹“C:\BLETest\app\libs\”，可以看到“acssmcio-x.y.z.aar”和“smartcardio-x.y.z.aar”文件。将这两个类库文件复制到项目文件夹“app\libs”。



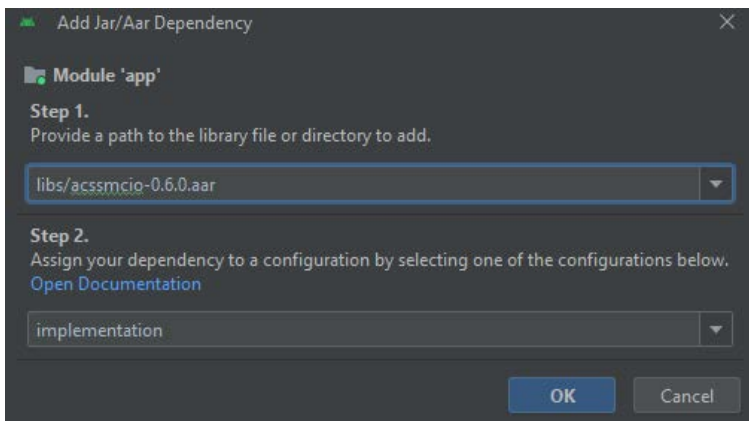
2) 导航到 File -> Project Structure -> Dependencies.



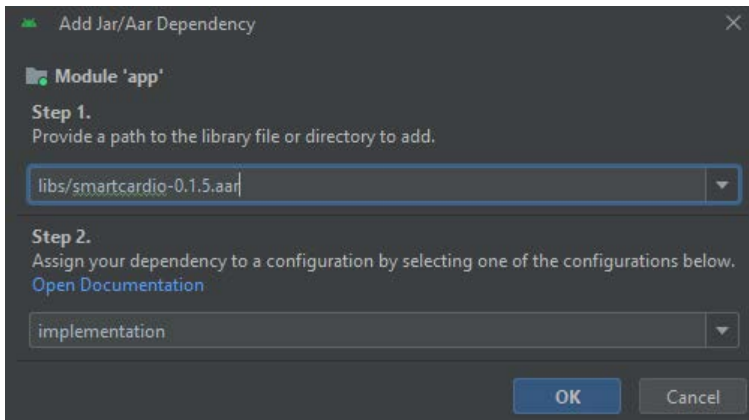
- 3) 在“Declared Dependencies”选项卡中，点击下拉菜单并选择“Jar Dependency”。

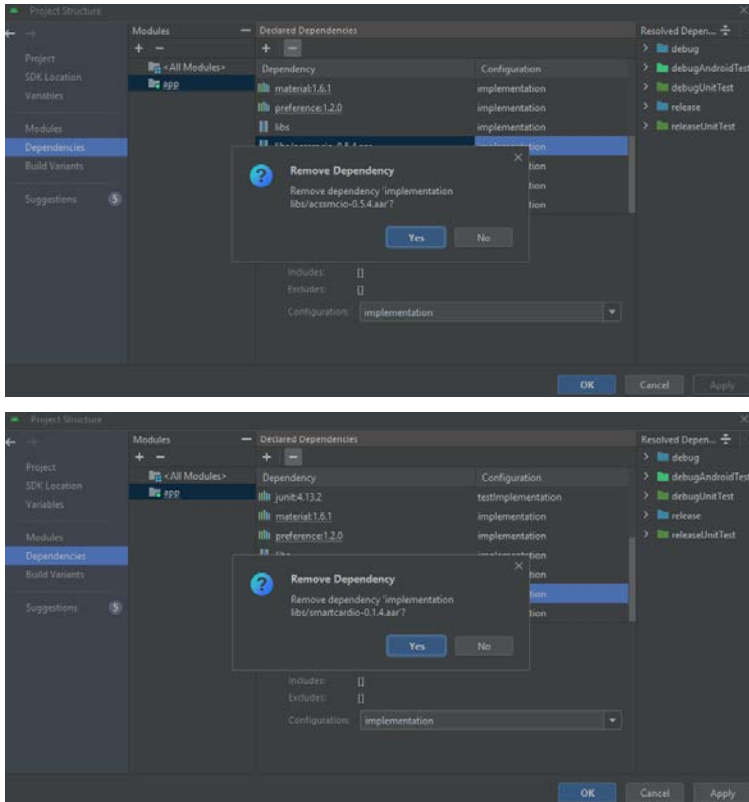


- 4) 在“Add Jar/Aar Dependency”对话框中，输入路径到“libs/acssmcio-x.y.z.aar”并选择“implementation”配置。

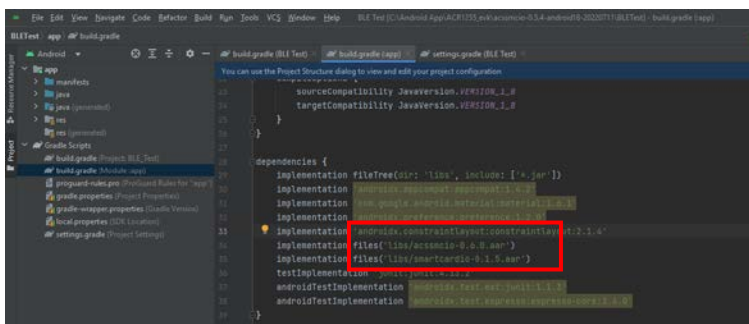


- 5) 按照上述步骤添加“libs/smartcardio-x.y.z.aar”。





6) 在您应用程序的“build.gradle”文件中，可以看到下列代码行。



implementation files('libs/acssmcio-x.y.z.aar')

implementation files('libs/smartcardio-x.y.z.aar')

7) 关于添加 ACR1555U 终端类型的说明

a. 找到 strings.xml 文件:

打开项目并导航到 res/values/目录，找到 strings.xml 文件。

b. 添加终端类型:

在 strings.xml 文件中，将下列代码行插入<string-array>标签，添加 ACR1555U 终端类型:

<item>ACR1555U</item>



```
32 <!-- Terminal types -->
33 <string name="select_terminal_type">Select a terminal type</string>
34 <string-array name="terminal_types_array">
35 <item>ACR3901U-S1/ACR3901T-W1</item>
36 <item>ACR1255U-J1</item>
37 <item>AMR220-C</item>
38 <item>ACR1255U-I1 V2</item>
39 <item>ACR1555U</item>
40 </string-array>
41
```

a) **搭建项目和测试:**

添加好终端类型后，继续搭建项目。搜索 ACR1555U 终端的功能将启用。

需要注意的是，更新库时可能出现不兼容问题，需要对应用程序做进一步的修改。



1.2. ACS 智能卡 I/O iOS 框架

“ACSSmartCardIO” 框架主要提供与 ACS 蓝牙读写器进行通信的类和接口。该框架基于 JSR 268 [1] 定义的 Java 智能卡 I/O API 中的服务提供商接口（TerminalFactorySpi）。

Java 智能卡 I/O API 定义了使用 ISO/IEC 7816-4 APDU 与智能卡进行通信的 Java API。该 API 允许 Java 应用程序与智能卡上运行的应用程序进行交互，实现卡片数据的存储和检索等功能。

“SmartCardIO” 框架包含下列从 OpenJDK 移植到 Swift 的源代码。

- javax.smartcardio
- java.security.Provider

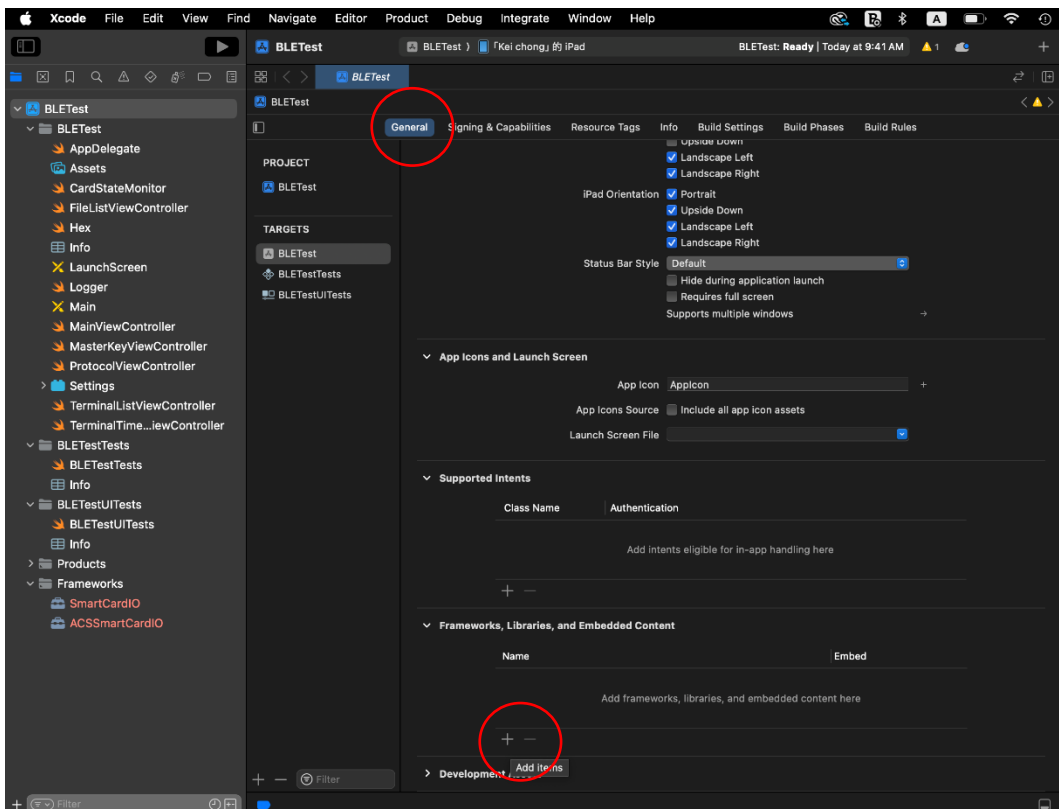
[1] <https://jcp.org/en/jsr/detail?id=268>

1.2.1.iOS 库的安装

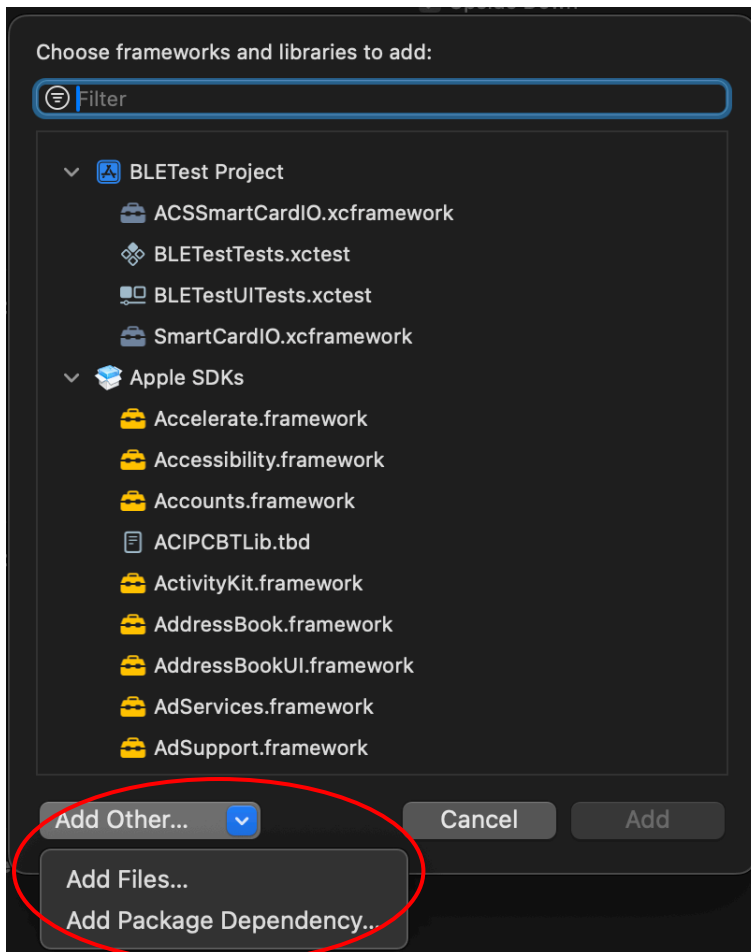
1. 要将该框架用于您的项目，请将“BLETest\SmartCardIO.xcframework”和“BLETest\ACSSmartCardIO.xcframework”这两个文件夹复制到您的项目文件夹。



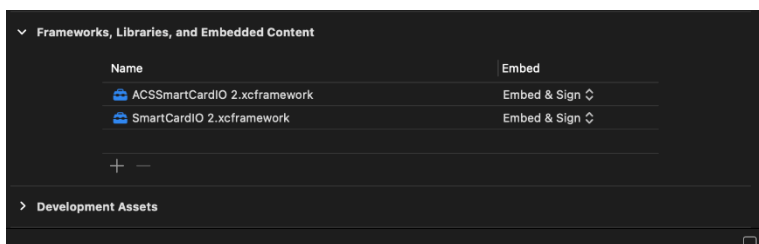
2. 在 Targets 的 General 标签页中，点击 Embedded Binaries 下的“+”号。



3. 在弹出的对话框中，点击“Add Other...”按钮，将框架添加到您的项目当中。



4. 您会看到下列库被添加到项目中



- 5) 关于添加 ACR1555U 终端类型的说明

找到 TerminalListViewController，并将下列代码添加到“ACR1255-J1 V2”小节的后面。

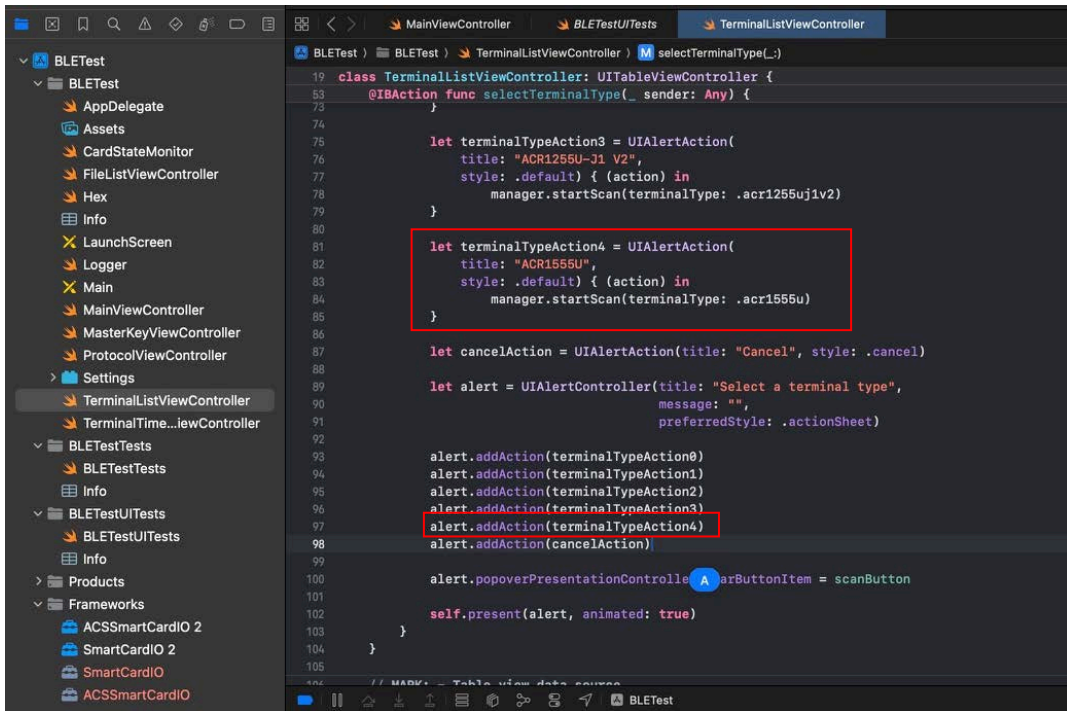
```

let terminalTypeAction4 = UIAlertAction{
    title: “ACR1555” ,
    style: .default) { (action) in
        manager.startScan(terminalType: .acr1555u)
    }

```

然后将下列代码添加到 alert.addAction(terminalTypeAction3)后面

```
alert.addAction(terminalTypeAction4)
```



6) 搭建项目和测试:

添加好终端类型后，继续搭建项目。搜索 ACR1555U 终端的功能将启用。

需要注意的是，更新库时可能出现不兼容问题，需要对应用程序做进一步的修改。