



**Advanced Card Systems Ltd.**  
Card & Reader Technologies

# ACS Bluetooth ライブラリ更新ガイド

## [iOS 及び Android]

ユーザーガイド V1.01



## 目次

1.0.	紹介.....	3
1.1.	ACSスマートカード I/O Android ライブラリ.....	3
1.1.1.	Android ライブラリのインストール.....	4
1.2.	ACSスマートカード I/O iOS フレームワーク.....	7
1.2.1.	iOS ライブラリのインストール.....	8



## 1.0. 紹介

このユーザーズガイドでは、お客様が作成したアプリケーションで ACS Bluetooth ライブラリを更新する方法について説明します。下記は Bluetooth ライブラリを更新する一般的な手順になります。ライブラリを更新する時に、互換性のない問題が発生する可能性があり、アプリケーションを修正する必要があることに注意してください。

## 1.1. ACS スマートカード I/O Android ライブラリ

“acssmcio”ライブラリには、主に ACS Bluetooth リーダー/ライターと通信するクラスとインタフェースがあります。このライブラリは、JSR 268[1]で定義されている Java スマートカード I/O API におけるサービスプロバイダインタフェース (TerminalFactorySpi) に基づいています。

Java スマートカード I/O API は、ISO/IEC 7816-4 APDU を介してスマートカードと通信する Java API を定義します。この API は、Java アプリケーションがスマートカードで動作するアプリケーションと対話することを可能にし、カードデータのメモリや検索などの機能を実現します。

“smartcardio”ライブラリが OpenJDK から Java パッケージをインポートします：

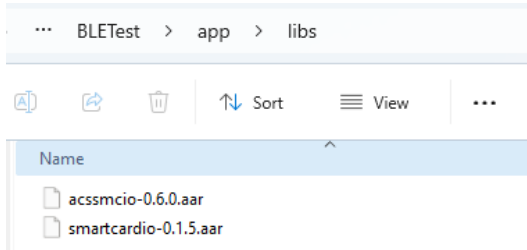
- javax.smartcardio
- sun.net.www
- sun.nio.cs
- sun.security.action
- sun.security.jca
- sun.security.util

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

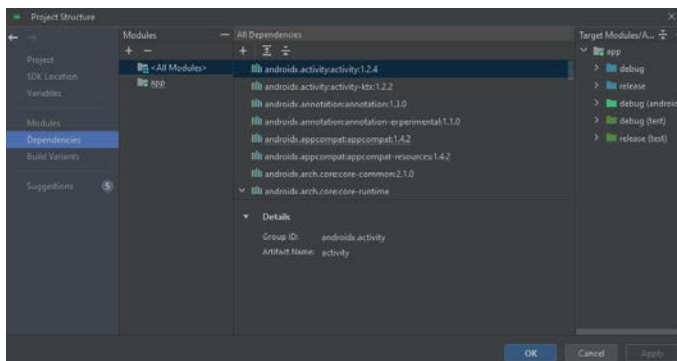
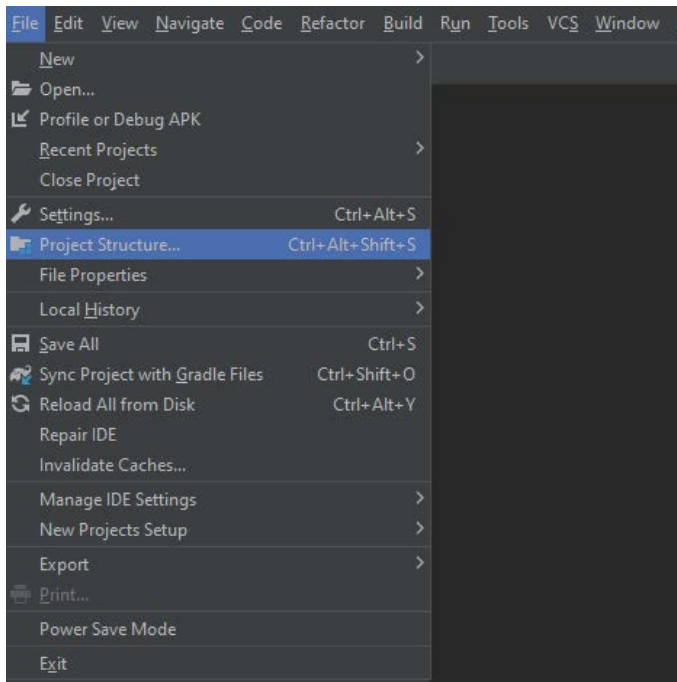
### 1.1.1. Android ライブラリのインストール

1) 更新された Android ライブラリをダウンロードして解凍します。

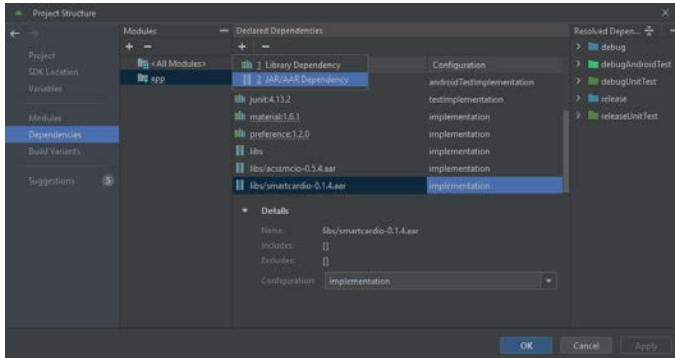
解凍後のクラスライブラリフォルダ“C:\BLETest\app\libs”を見つけると、“acssmcio-x.y.z.aar”と“smartcardio-x.y.z.aar”ファイルが表示されます。2つのクラスライブラリファイルをプロジェクトフォルダ“app\libs”にコピーします。



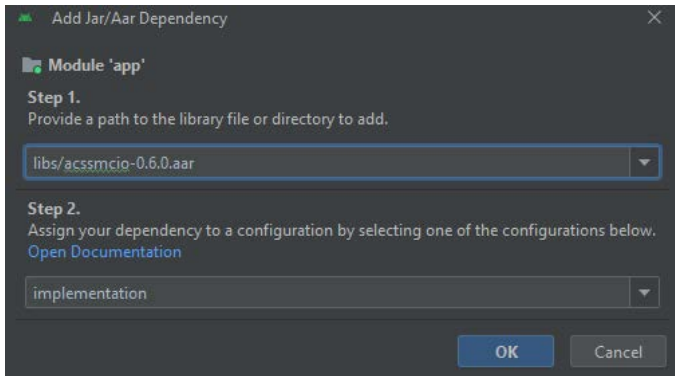
2) File -> Project Structure -> Dependencies に移動します。



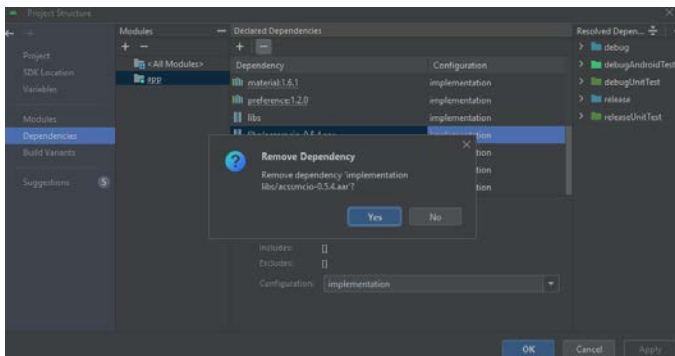
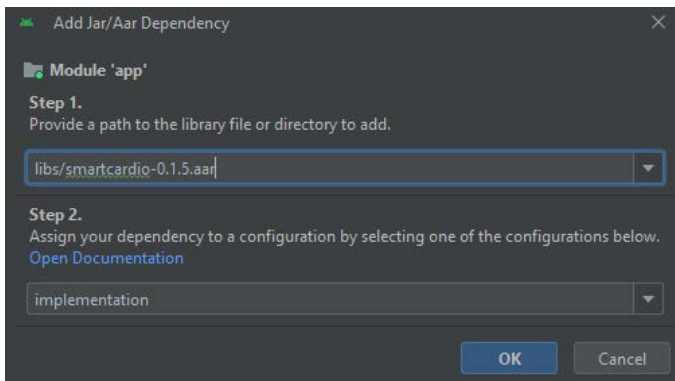
- 3) “Declared Dependencies”タブで、ドロップダウンメニューをクリックして“Jar Dependency”を選択します。

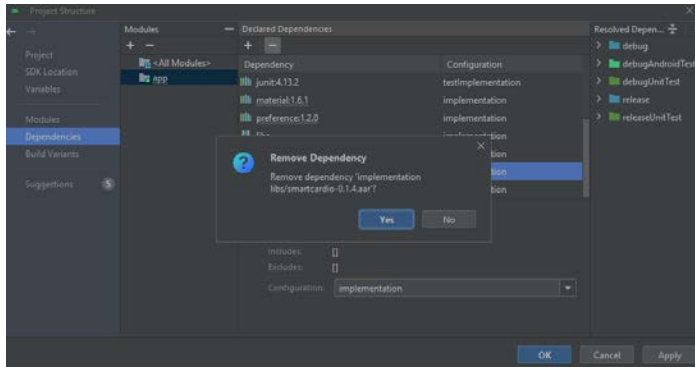


- 4) “Add Jar/Aar Dependency”ダイアログで、“libs/acssmcio-x.y.z.aar”にパスを入力し、“implementation”構成を選択します。

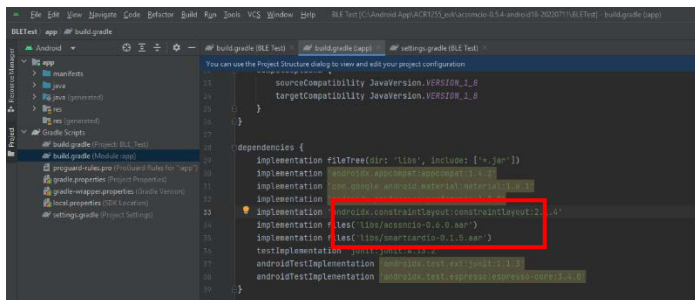


- 5) 上記手順に従って“libs/smartcardio-x.y.z.aar”を追加します。





6) アプリケーションの“build.gradle”ファイルには、次のコード行が表示されます。



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

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

7) ACR1555U 端末タイプの追加について

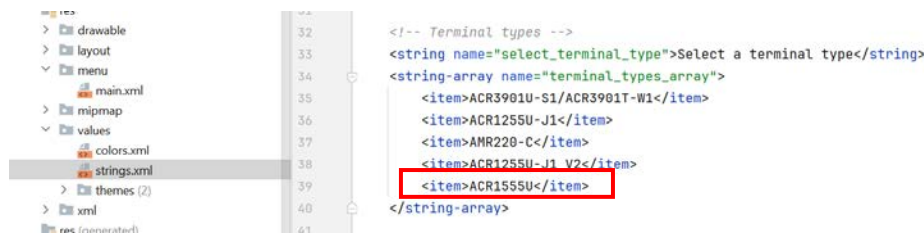
a. strings.xml ファイルを見つける :

プロジェクトを開き、res/values/ディレクトリに移動して、strings.xml ファイルを見つけます。

b. 端末タイプを追加します :

strings.xml ファイルで、次のコード行を <string-array> タブに挿入し、ACR1555U ターミナルタイプを追加します。

<item>ACR1555U</item>



a) プロジェクト構築とテスト :

ターミナルタイプを追加したら、プロジェクトを構築し続けます。ACR1555U 端末を検索する機能が有効になります。

ライブラリを更新する時に、互換性のない問題が発生する可能性があり、アプリケーションを修正する必要があることに注意してください。



## 1.2. ACS スマートカード I/O iOS フレームワーク

“ACSSmartCardIO”ワークには、主に ACS Bluetooth リーダー/ライターと通信するクラスとインタフェースがあります。このワークは、JSR 268[1]で定義されている Java スマートカード I/O API におけるサービスプロバイダインタフェース (TerminalFactorySpi) に基づいています。

Java スマートカード I/O API は、ISO/IEC 7816-4 APDU を使用してスマートカードと通信する Java API を定義します。この API は、Java アプリケーションがスマートカードで動作するアプリケーションと対話することを可能にし、カードデータのメモリや検索などの機能を実現します。

“SmartCardIO”フレームワークには、OpenJDK から Swift に移行する次のソースコードが含まれています。

- javax.smartcardio
- java.security.Provider

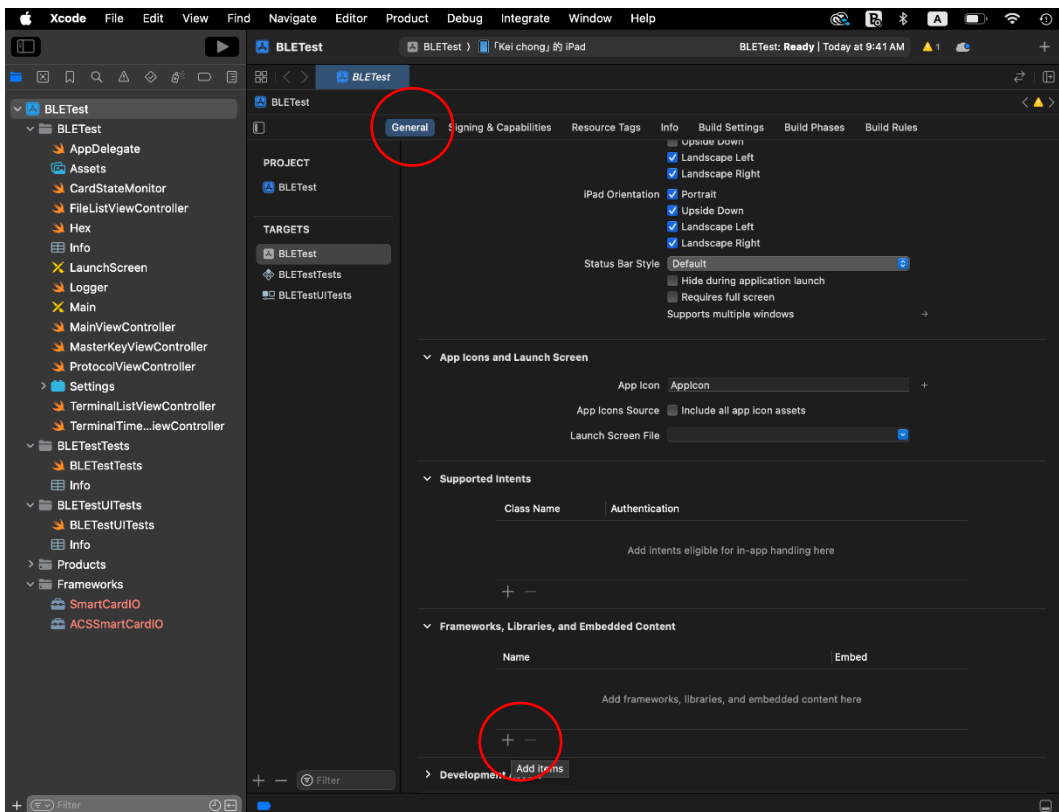
[1] <https://jcp.org/en/jsr/detail?id=268>

## 1.2.1. iOS ライブラリのインストール

1. フレームワークをプロジェクトに使用するには、“BLETest\SmartCardIO.xcframework”と“BLETest\ACSSmartCardIO.xcframework”の 2 つのフォルダをプロジェクトフォルダにコピーしてください。

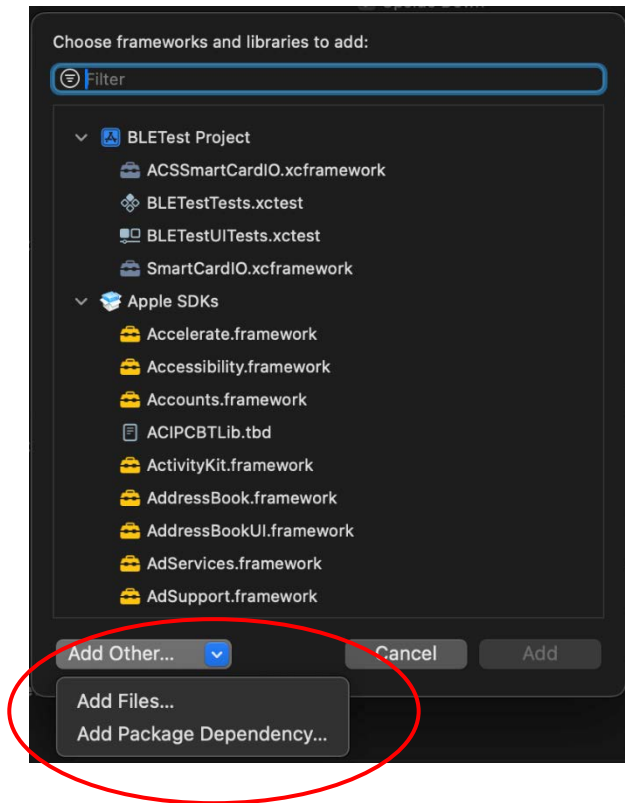


2. Targets の General タブページで、Embedded Binaries の下の“+”をクリックします。

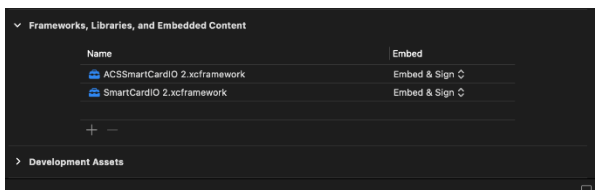




3. ポップアップダイアログボックスで、“Add Other...”ボタンをクリックして、フレームをプロジェクトに追加します。



4. プロジェクトに追加された次のライブラリが表示されます



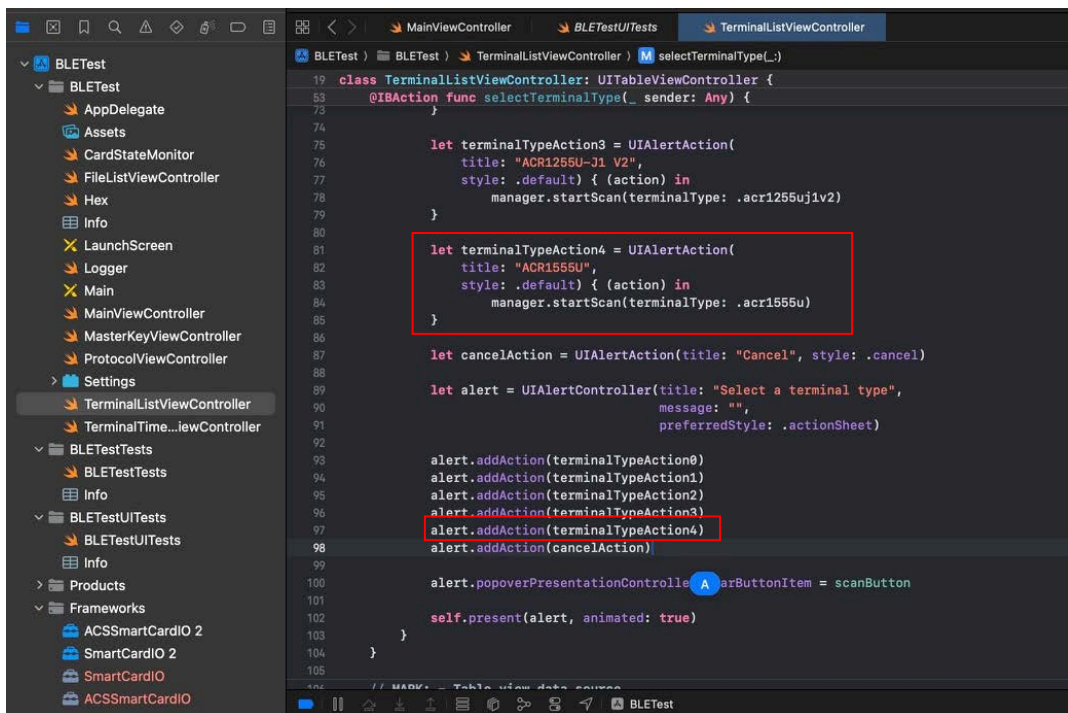
### 5) ACR1555U 端末タイプの追加について

TerminalListViewController を見つけ、次のコードを“ACR1255-J1 V2”セクションの後ろに追加します。

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

そして alert.addAction (terminalTypeAction 3) の後に次のコードを追加します

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



```

class TerminalListViewController: UITableViewController {
    @IBAction func selectTerminalType(_ sender: Any) {
        let terminalTypeAction3 = UIAlertAction(
            title: "ACR1255U-J1 V2",
            style: .default) { (action) in
                manager.startScan(terminalType: .acr1255uj1v2)
            }

        let terminalTypeAction4 = UIAlertAction(
            title: "ACR1555U",
            style: .default) { (action) in
                manager.startScan(terminalType: .acr1555u)
            }

        let cancelAction = UIAlertAction(title: "Cancel", style: .cancel)

        let alert = UIAlertController(title: "Select a terminal type",
            message: "",
            preferredStyle: .actionSheet)

        alert.addAction(terminalTypeAction0)
        alert.addAction(terminalTypeAction1)
        alert.addAction(terminalTypeAction2)
        alert.addAction(terminalTypeAction3)
        alert.addAction(terminalTypeAction4)
        alert.addAction(cancelAction)

        alert.popoverPresentationController?.barButtonItem = scanButton

        self.present(alert, animated: true)
    }
}

```

## 6) プロジェクト構築とテスト:

ターミナルタイプを追加したら、プロジェクトを構築し続けます。ACR1555U 端末を検索する機能が有効になります。

ライブラリを更新する時に、互換性のない問題が発生する可能性があり、アプリケーションを修正する必要があることに注意してください。