

# 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

前に告知なく変更される場合があります



# 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

Page 2 of 10



### 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

Page 3 of 10



### 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。





Page 4 of 10



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。

-	Project Structure				×
+	Project Structure Project Coll Location Variable Dependencies Build Variants Suggestions	Modules —	Declared Dependence + - III 1 Linary Dependency III 2 JAK/ARD Dependency IIII 2 JAK/ARD Dependency III 2 JAK/ARD Depe	Cardiguation Addical Technolomentation tectinoplementation implementation implementation implementation implementation	× RestAred Depend. ÷ = > in debug > in debughedeadTest > in debughedeatTest > in release > int releasetSeatTest

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。

🛎 🛛 Add Jar/Aar Dependency		×
📑 Module 'app'		
<b>Step 1.</b> Provide a path to the library file or directory to a	add.	
libs/acssmcio-0.6.0.aar		
Step 2. Assign your dependency to a configuration by s Open Documentation	selecting one of the configuratio	ins below.
implementation		
	ок	Cancel

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



Page 5 of 10





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。

🛎 Elle Edit View Navigate Code Befactor Build	Run Jools VCS Window Hitp REFTest (C\Android App\ACR1255_edAassencio-0.5.4 android18-20220711\8ETest -build.gradie (app)	
BLETest app 🛹 build gradie		
ु 🛎 Android 👻 😳 🗄 🗢 —	🔎 build gradie (8LE Test) = 🖉 build gradie (app) 👘 🔎 settings gradie (8LE Test) 🚿	
A Bit way     A Bit way	<pre>Years use the Properticular data is non on it diagram projectioning and an source Comparison of the Properticular State S</pre>	

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 ファイルを見つけて:

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>

== 123	10 A.	
> 🛅 drawable	32	Terminal types
> 🛅 layout	33	<string name="select_terminal_type">Select a terminal type</string>
Y Di menu	34 😇	<string-array name="terminal_types_array"></string-array>
🚔 main.xml	35	<item>ACR3901U-S1/ACR3901T-W1</item>
> 🛅 mipmap	36	<item>ACR1255U-J1</item>
values	37	<item>AMR220-C</item>
strings.xml	38	<pre><item>ACR1255U-J1 V2</item></pre>
> 🖿 themes (2)	39	<item>ACR1555U</item>
> 🛅 xml	40 🖨	
The res (concentrated)	2.2	

#### 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。

Page 6 of 10



## 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。

< > BLETest		<u> </u>		⊙ •	Q
名稱	修改日期			種類	
ACSSmartCardIO.xcframework	今日上午9:27			資料夾	
BLETest	今日 上午 9:27			資料夾	
BLETest.xcodeproj	18/11/2021 下午4:56		29 KB	Xcode Projec	
🗎 BLETestTests	今日 上午 9:27			資料夾	
BLETestUlTests	今日 上午 9:27			資料夾	
SmartCardIO.xcframework	今日 上午9:27			資料夾	

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

🖨 Xcode File Edit View Fin	d Navigate Editor	Product	Debug	Integrate	Window	Help		🗟 🖪 🖇	Α	Ì	()	•
	📕 BLETest	🛛 BLE	Test 🕽 📘 🛙	Kei chong」的	) iPad		BLETest: Ready	foday at 9:41 AM	<b>A</b> 1	<b>@</b>		
	🔠 I < 🚿 🔼 BLETE	est										
V 🖪 BLETest	🖾 BLETest											▲ >
V BLETest	D	General	signing &	Capabilities	Resource Ta	igs li	Info Build Settings Build Pha	ises Build Rule	s			
🔌 AppDelegate 🚾 Assets	PROJECT						<ul> <li>Upside Down</li> <li>✓ Landscape Left</li> <li>✓ Landscape Right</li> </ul>					
CardStateMonitor FileListViewController	🚨 BLETest				iPad Oriei	ntation	<ul> <li>Portrait</li> <li>Upside Down</li> </ul>					
🎽 Hex	TARGETS						Landscape Left					
Info	BLETest				Statue Ba	ar Style	Default					
LaunchScreen	BLETestTests				010105 00	a otyle	Hide during application launch					
× Main	District BLETestUITests						Requires full screen					
MainViewController							Supports multiple windows					
🌙 MasterKeyViewController												
🌂 ProtocolViewController		Ť,	App icons a	na Launch S	creen							
> 👛 Settings					Ap	op Icon	Applcon					
TerminalListViewController					App Icons	Source	🔲 Include all app icon assets					
TerminalTimeiewController					Launch Scre	en File		2				
BLE lest lests												
			Supported I	Intents								
✓ ■ BLETestUITests				Class Name	Aut	henticat	tion					
🔌 BLETestUITests												
🖽 Info						Add inte	ents eligible for in-app handling he					
> 📰 Products												
Frameworks												
SmartCardIO												
		~ 1	Framework	s, Libraries, a	and Embedde	d Conte	ent					
				Name				Embed				
							orks, libraries, and embedded conte					
	+ - 🗑 Filter	<b>→</b>	Developme	Add items								
+ 🖘 Filter												



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

✓ Framework	s, Libraries, and Embedded Content		
	Name	Embed	
	ACSSmartCardIO 2.xcframework	Embed & Sign 🗘	
	🚘 SmartCardIO 2.xcframework	Embed & Sign 🗘	
> Developme	nt Assets		

#### 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)

Page 9 of 10





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。

Page 10 of 10